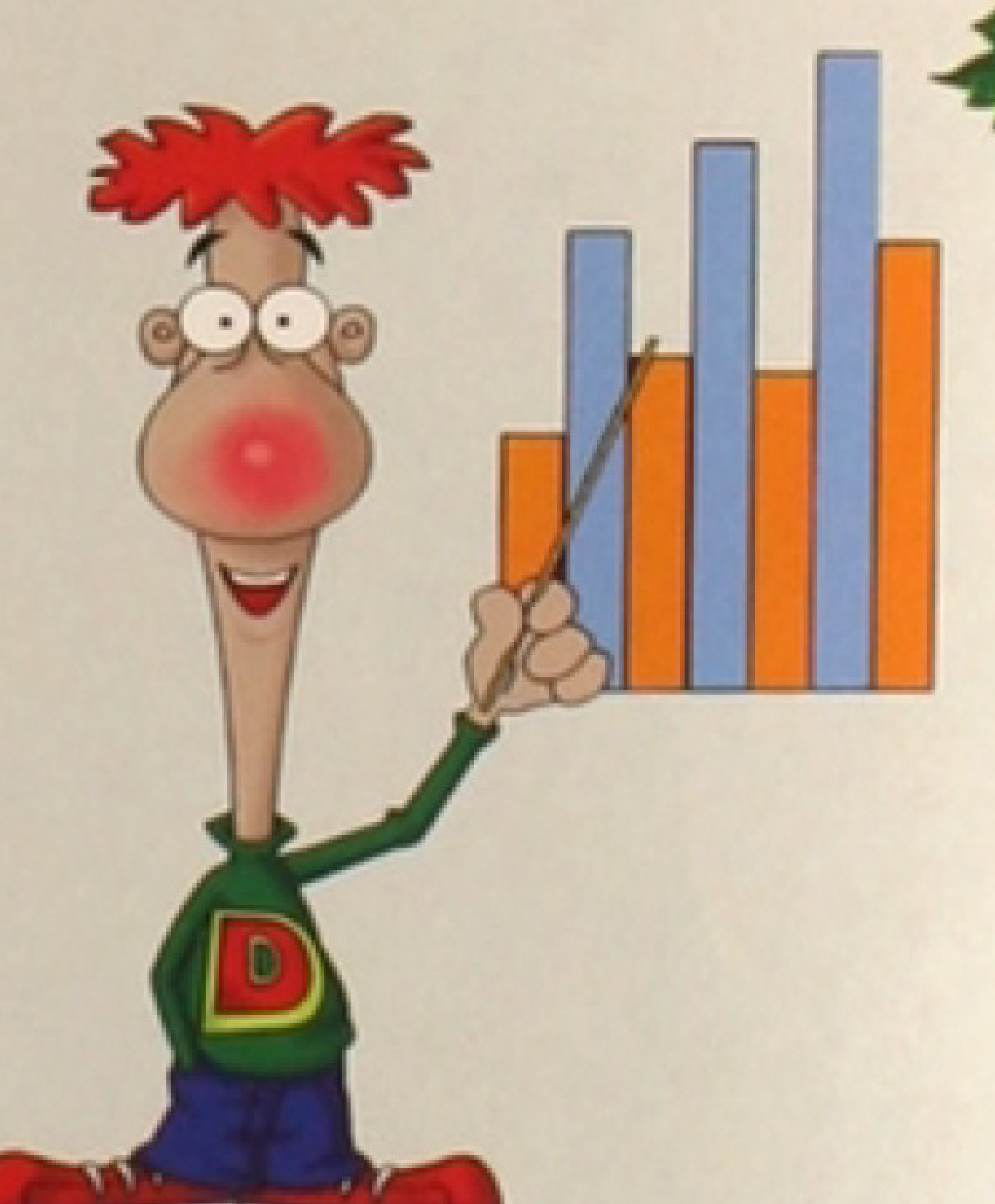
管理心理学

Management Psychology

主编　刘永芳





清华大学出版社

教育部人文社会科学一般项目成果

管理心理学

刘永芳　主编

清华大学出版社

北　京

内 容 简 介

本书作为一本针对广大高校师生、人力资源管理从业人员以及心理学爱好者的专业教材，在对管理心理学领域基础理论、基本思想进行充分阐述的同时，尤其注重教材的可用性、灵活性、实践性。全书分为总论、动机管理、认知管理、情绪管理、行为管理、组织管理等六篇共十八章。在体例安排上，每章结尾部分都安排有“本章小结”、“本章思考题”，有的章节后面还有“案例分析”、“技巧库”和“小测验”，此外，部分章节内安排有“管理小贴士”、“阅读材料”等内容，以开阔读者的视野，加深其对相关知识和原理的理解。在内容安排上，囊括了管理心理学的基本范畴、研究方法、基础知识、理论框架，并对近年来管理心理学的研究重点做了深入、细致的探讨，增添了许多新概念、新理论、新方法，体现了当今国内外管理心理学的重要理论成果，有助于读者更好地把握管理心理学界的前沿动态。

本书旨在为读者打开一扇窗，展示一片管理心理学的广阔天地，为高等院校管理学、心理学等专业的师生以及从事人力资源管理的人员提供帮助。

图书在版编目（CIP）数据

管理心理学/刘永芳主编. —北京：清华大学出版社，2008.1

ISBN 978-7-302-16622-1

Ⅰ. 管…　Ⅱ. 刘…　Ⅲ. 管理心理学　Ⅳ. C93-05

中国版本图书馆CIP数据核字（2007）第195223号

责任编辑：吴颖华（wuyinghua2003@163.com）
封面设计：张　岩
版式设计：高　伟
责任校对：姜　彦
责任印制：
出版发行：清华大学出版社
http://www.tup.com.cn
c-service@tup.tsinghua.edu.cn
社 总 机：010-62770175
投稿咨询：010-62772015
地　　址：北京清华大学学研大厦A座
邮　　编：100084
邮购热线：010-62786544
客户服务：010-62776969
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185×260　**印　　张：**22　**字　　数：**488千字
版　　次：2008年1月第1版　**印　　次：**2008年1月第1次印刷
印　　数：1～5000
定　　价：32.80元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题，请与清华大学出版社出版部联系调换。联系电话：010-62770177转3103　　产品编号：021231-01

前言

PREFACE

管理心理学是研究管理过程中个体的行为规律及其潜在心理机制的一门学科。作为应用心理学的一个重要分支，管理心理学自20世纪中叶发展为一门独立的学科以来，经过半个多世纪的不断壮大，如今已经成为备受心理学、管理学领域研究者及管理实践者们关注的一门学科。

理论的发展和实践的需要带来学科的繁荣。近些年来，国内外各种名目的管理心理学著作可谓卷帙浩繁、层出不穷。每本书特色各异，往往在某一个或某几个方面显露出闪光点，但观其大略不难发现，这些著作在结构上多遵循“个体心理——群体心理——组织心理”三层次的架构，仅在内容上作了顺序和详略的调整。在这种架构下，个体在不同层面（个体层面、群体层面、组织层面）上的交互作用是研究的重点，为组织管理服务是研究的目标。因此，与其将它笼统地称为管理心理学，不如称为“管理学取向的管理心理学”或者“从组织角度出发的管理心理学”。作为应用心理学的一门分支学科，如果管理心理学不能充分体现心理学特色的话，则一方面管理学研究者不能从中汲取心理学的营养，另一方面对于广大心理学工作者来说也是一种遗憾。鉴于这种情况，我们提出了构建“心理学取向的管理心理学”的设想，该设想大体上包括两层意思：（1）突出人及其心理发展本身作为组织管理目标之一的地位，在重视管理心理学作为工具性学科为实现组织目标服务的同时，强调其作为非工具性学科为实现组织中人的全面发展目标服务的属性，体现心理学的人文关怀、个体关心、科学主义取向等特征；（2）将企业组织看作背景和舞台，而把知、情、意、行等心理活动作为在前台演戏的主角，依照心理学的逻辑来统驭相关的知识和内容，形成以动机管理、认知管理、情绪管理和行为管理为主线的管理心理学体系。以该思路为主导构建管理心理学体系，既可以突出心理学的特色和优势，又能更加深入细致地揭示组织行为背后的微观心理机制，更好地体现心理学的独特作用，为实现平等的、互动的、无创伤的人本管理提供强有力的知识和理论支持。

近三十年来，心理学、管理学及其他相关学科的迅猛发展，为人们打开了窥探管理世界奥秘的新视野。新理论、新观点、新知识、新材料的不断涌现让我们有了一种强烈的冲动：将这些新鲜血液及时吸收进来，一方面为我所用，丰满管理心理学学科体系，增强学科的生命力和活力；另一方面将这些思想的瑰宝从散碎的文献中发掘出来，予以提取整合，对于理论本身的发展和完备也大有裨益。在本书中，我们力求融会近几年涌现出来的新理论、

新观点、新知识、新材料，将其揉进我们的体系中，既反映当今管理心理学的前沿动向，又不失为一个完整的理论体系。

本书是一本著作，也是一本教科书。语言的简洁明了，行文的系统逻辑，内容结构的合理严谨是我们遵循的准则。考虑到教学的需要，本书每章结尾都安排有“本章小结”、“本章思考题”，有的章节后面还有“案例分析”、“技巧库”和“小测验”。此外，部分章节内安排有“管理小贴士”、“阅读材料”等内容，以开阔学生的视野，加深其对相关知识和原理的理解。可以说，本书是心理学、管理学研究者的参考书，是高等院校学生的教科书，还是管理实践者的指导书。

本书是集体协作完成的一项科研成果，负责各章编写的人员如下：第 1 章，刘永芳、王怀勇；第 2、3 章，赵立军、刘永芳；第 4 章，王怀勇、刘永芳；第 5、11、12 章，方学梅；第 6、10 章，许科；第 7、9 章，王彦；第 8 章，王大伟；第 13、14 章，卢会志；第 15 章，庄锦英；第 16、17 章，房慧聪；第 18 章，王志勇、刘永芳。书稿完成后，主编对全书进行了认真而艰苦的审阅和修改。

本书撰写过程中，参考并引用了国内外专家的大量研究成果，多数已开列在书后的参考文献中，有的在文中注明了出处，在此对所有文献的作者表示衷心感谢！清华大学出版社吴颖华编辑为本书的出版付出了大量心血和精力，在此表示深深的谢意！我的博士研究生许科在制订编写提纲和稿件审阅阶段帮我做了大量的具体工作，她的聪明能干给我留下了深刻的印象，谢谢她所付出的辛勤劳动！

由于编者水平有限，书中的疏漏和错误之处在所难免，恳请专家、学者和广大读者批评指正！

刘永芳

2007 年 7 月

目录

CONTENTS

第一篇　总论

第二篇　动机管理

第三篇　认知管理

第四篇 情绪管理

第五篇　行为管理



第一篇　总论

第一章　管理心理学概述

管理心理学在西方又称为工业与组织心理学，是研究组织管理活动中人的行为规律及其潜在心理机制的一门学科。它属于心理学的研究范畴，是心理学的一个重要分支。近年来，随着理论的不断完善与应用范围的逐步扩大，管理心理学已经发展成为一门独立的学科，在我国社会主义市场经济与现代化建设中扮演着越来越重要的角色。本章将对管理心理学的研究对象、研究内容、研究逻辑及研究方法等基本问题进行探讨，以便为后面的章节奠定基础。

第一节　管理心理学的研究对象和内容

一、管理心理学的研究对象

什么是管理心理学？不同的教材和专著从不同的角度给出了各自的定义，这些定义尽管在内容上有所区别，但就其实质而言则是相同的。一般说来，管理心理学是研究组织管理活动中人的行为规律及其潜在的心理机制，并用科学的方法改进管理工作，不断提高工作效率与管理效能，最终实现组织目标与个人全面发展的一门学科。这个定义主要有以下三层含义：

第一，管理心理学的研究对象是人的行为规律及其潜在的心理机制。也就是说，管理心理学既探索组织管理活动中人的行为规律，又揭示这些行为背后潜在的心理机制。它把人的行为规律及其心理机制结合起来进行研究，其原因在于人的行为与心理之间具有密不可分的关系。一方面，行为是在一定心理活动的指导下进行的，它是心理活动的外在表现，要想真正了解人的行为，就要探索其背后潜在的心理机制；另一方面，心理又是行为的内在动因，是调节、控制行为的内部过程，要想揭示人类心理的奥秘，就必须分析人的外在行为。由此可见，管理过程中人的行为规律及其心理机制都是管理心理学所要探讨的对象。

第二，管理心理学的研究范围是组织管理活动中人的行为规律及其潜在的心理机制。换句话说，管理心理学不是探索一般情境下人的行为规律及其潜在的心理机制，而是把研究范围严格限定在组织管理活动这一特定情境中，这里的“组织”不仅包括企业组织，还涵盖政治、经济、军事、教育等社会组织。

第三，管理心理学研究的目的是在掌握组织管理活动中人的行为规律及其潜在心理机制的基础上，运用科学的方法改进管理，充分调动人的积极主动性，不断提高工作效率与管理效能，最终实现组织目标与个人的全面发展。具体地说，管理心理学是一门应用性较

强的学科，它来源于社会管理实践，也服务于社会管理实践。管理心理学的研究就是要进一步提高组织管理的科学化水平，充分挖掘人的发展潜力，以实现组织和个人的发展目标。

二、管理心理学的研究内容

管理心理学的研究对象决定了其研究内容应基于人的心理活动而展开，并辐射到行为和组织层面的相关问题。基于此种考虑，本书将以组织情境中人的心理活动为主线来安排相关内容，旨在突出管理心理学的心理学特色，增强其心理学色彩。除了“总论”部分（第一篇）介绍管理心理学基本问题的三章内容之外，其余五篇内容将按顺序分别讨论动机管理、认知管理、情绪管理、行为管理和组织管理等主题。下面分别对这五个主题做一简要说明。

（一）动机管理

动机是其他一切心理活动的起点和前提。在现代管理活动中，如何有效激发员工的动机，充分调动员工的工作积极性、主动性和创造性，成为各类组织提高工作绩效、实现发展目标的关键。所以，动机管理问题亦应该成为管理心理学首先探讨和解决的问题。基于这种考虑，“总论”之后，本书的第二篇即以“动机管理”为主题，用三章内容分别讨论动机的一般概念和规律、激励理论和激励理论的应用等。

（二）认知管理

认知是人们获得或应用知识的过程，是最基本的心理过程。现代组织管理强调以人为中心，因此要想富有成效地开展管理工作，就必须了解组织情景中人的认知加工过程及其规律，了解得愈透彻，愈能对症下药，做好管理工作。本书第三篇以“认知管理”为主题展开讨论，用三章内容分别探讨组织中的知觉与印象管理、组织中的记忆与学习、组织中的思维与决策等。

（三）情绪管理

随着生活节奏的加快及竞争的日趋激烈，组织及组织中的个人面对的应激事件越来越多，遭遇挫折的机会越来越高，压力越来越大，情绪管理问题在日常的组织管理活动中显得越来越重要。这方面的研究不仅有助于降低员工的压力，提高员工的身心健康水平，而且有助于缓解组织冲突，调动员工的工作积极性，实现组织的发展目标。本书的第四篇以“情绪管理”为主题展开讨论，用三章内容分别探讨情绪劳动与管理、应激管理、员工卷入与管理等。

（四）行为管理

良好的行为是一切心理管理的最终目标。而组织情境中起主导作用的行为是领导行为。领导是管理工作的主角，不同的领导行为会造成不同的组织心理气氛，给人以不同的心理影响，而这又是影响员工行为表现的最重要的条件。有效的领导行为有助于领导者作出正确决策，实现组织目标及改善领导者与被领导者之间的关系。因此，在以“行为管理”为

主题的第五篇中，用三章内容着重探讨领导行为及其相关问题，内容涉及领导的一般概念及领导有效性的理论、领导策略与方法、领导者的选拔与培养等。

（五）组织管理

组织是为了达到一定的目标，由不同结构、不同层次、不同职能的群体组合而成的系统。管理心理学研究的主体是组织中的主体，研究的活动是在组织背景下展开的，研究的目标最终也要落实到组织的发展上。组织是团体或个体实现某种目标的舞台，整个组织的状况直接影响群体或个体的行为效率。因此，除了研究个体层面的心理和行为问题之外，管理心理学还需要研究由个体形成的组织的特征及组织层面上的群体心理和行为规律，这对于如何更好地调动组织成员的积极性，充分利用人力、物力、财力，提高组织的效率是十分重要的。本书的第六篇以“组织管理”为主题，用三章内容分别探讨组织的一般概念和理论、组织士气和气氛、组织变革与发展等问题，内容涉及组织结构、组织文化、组织权力与政治、组织沟通与冲突、组织变革与发展等。

上面关于管理心理学研究内容的介绍及每个主题将会涉及的问题或内容，可以用图 1.1 加以概括和说明。

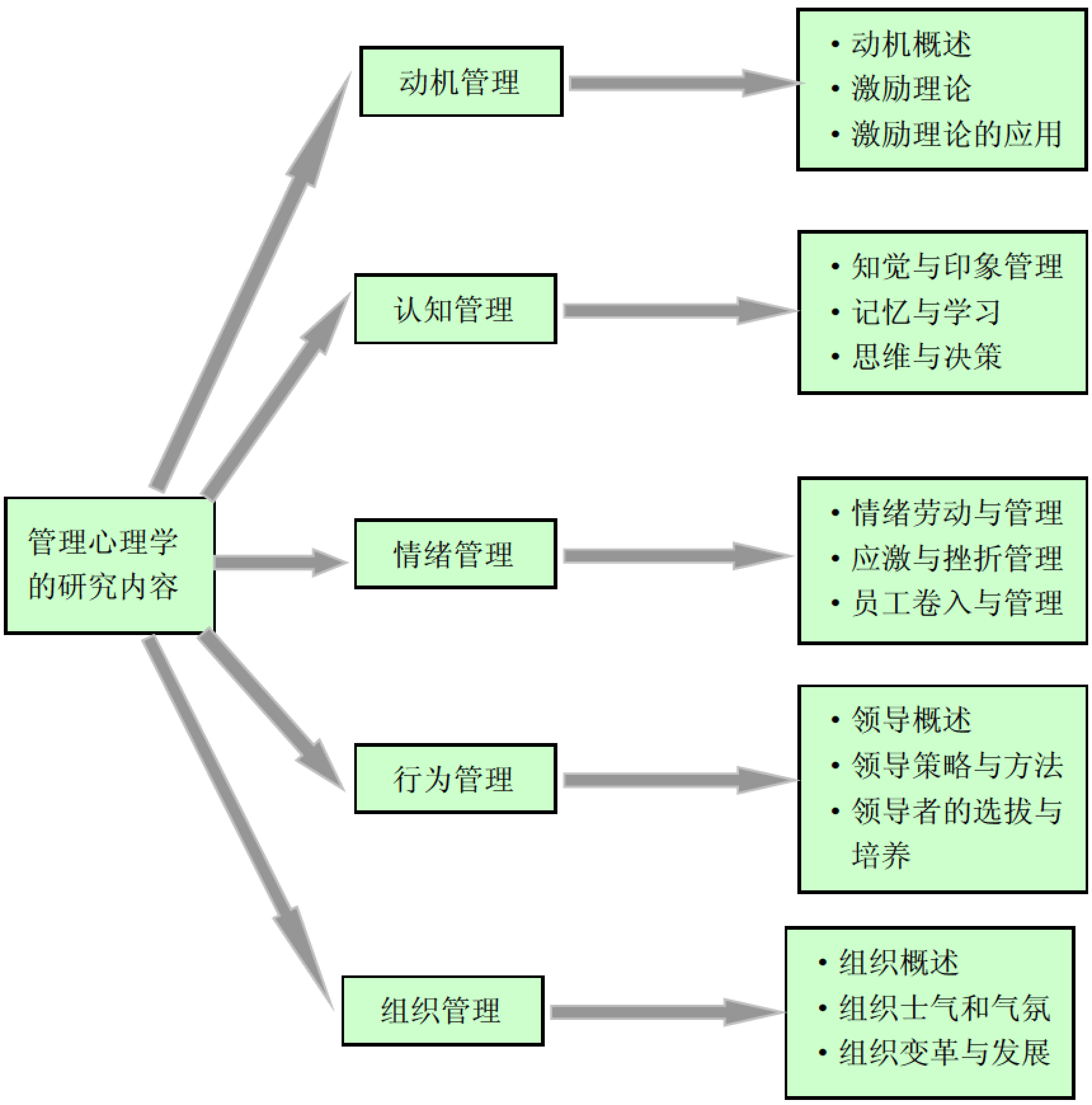


图 1.1　管理心理学的研究内容

第二节　管理心理学的学科性质和研究逻辑

一、管理心理学的学科性质

管理心理学是介于管理科学与心理科学之间的一门边缘性、交叉性学科，也与其他众多学科有密切关系，是一门综合性很强的学科。同时，管理心理学与管理实践紧密联系，其研究素材来源于管理实践，研究成果又要回到管理实践中去，是一门应用性很强的学科。管理心理学既有自然科学的性质，又有社会科学的性质。下面分别简要分析一下管理心理学与心理学、管理学、组织行为学等几个邻近学科的关系，以便读者更加明确这门学科的性质。

（一）管理心理学与心理学

心理学是研究人的心理现象及行为规律的一门科学，它研究的内容是人的一般心理过程和个性心理。管理心理学研究的则是组织管理活动中人的行为规律及其潜在的心理机制。心理学似一颗大树，它有许多分支，而管理心理学是其中的分支之一。通常，人们把管理心理学归入应用心理学的范畴，如在国际心理学会议上，往往将管理心理学放在应用心理学主题范围内来安排报告。管理心理学就是探讨如何把心理学的基本原理和研究成果应用到组织管理领域中去的一门应用学科。随着心理学基本理论研究不断深入，新成果不断涌现，管理心理学将得到进一步的发展；而管理心理学的发展，又将反过来验证和丰富心理学的基本理论，促进心理学的繁荣。由此可见，管理心理学与心理学是个别与一般、特殊和普遍的关系，二者密不可分但又不可互相替代。

（二）管理心理学与管理学

管理学是研究管理活动过程及其规律的科学，它是由一系列管理理论、职能、原则、方法等组成的科学体系，是社会科学、自然科学和技术科学相互渗透而形成的一门综合性学科。而管理心理学是研究管理活动过程中人的行为规律及其潜在心理机制的学科，也就是说，它主要研究管理过程中的心理问题。如果说管理学比较注重较为宏观的组织层面的问题的话，那么管理心理学则更注重较为微观的个体层面的问题；如果说管理学比较重视外在的行为层面的问题的话，那么管理心理学则更注重内在的心理层面的问题。所以，管理心理学与管理学的关系可以概括为：管理心理学是管理学的一个重要组成部分，是管理学的补充和发展。

（三）管理心理学与组织行为学

管理心理学与组织行为学这两门学科之间既有十分密切的联系，又有一定的区别，具

体分析如下：

1．管理心理学与组织行为学的区别

（1）研究的侧重点不同

管理心理学着重研究行为背后潜在的心理活动规律，侧重于把心理学的原理应用于组织管理活动中；而组织行为学则重点探讨行为特点和规律本身，把人的外显行为作为研究对象，以达到预测和控制行为的目的。管理心理学侧重于本源学的研究，组织行为学则侧重于现象学的研究。

（2）理论基础不同

管理心理学作为心理学的一个重要分支学科，它的理论源泉主要是心理学；而组织行为学作为行为科学的一个分支，它的理论来源更加多样化，不仅来自心理学，还来自管理学、社会学、人类学、政治学、经济学等学科。

（3）形成背景不同

管理心理学的形成经历了长期的理论和实践准备。1912 年，美籍德国心理学家闵斯特伯格（H.Muensterberg）出版了《心理学与工作效率》一书，首先正式把心理学应用到工业管理中。之后，随着心理学理论的不断发展及应用范围的逐步扩大，1958 年，美国心理学家利维特（H.J.Leavitt）正式提出“管理心理学”这一术语，并出版了第一本《管理心理学》著作，使管理心理学成为一门独立的学科。

组织行为学是由行为科学发展而来的，是行为科学与组织管理相结合而形成的分支学科。自 20 世纪 60 年代起，美国的工业和组织心理学研究开始从各大学的心理学系转到管理学院和研究生部。同时，这一领域的研究队伍除心理学家外，社会学家、人类学家甚至语言学家、数学家等陆续参加进来，知识来源不断扩大，形成一个跨众多学科的研究领域。

2．管理心理学与组织行为学的联系

（1）心理与行为密切相关

尽管管理心理学与组织行为学研究的侧重点不同，但事实上人的心理与行为之间是密不可分的。一方面，行为是心理的外在表现，组织行为学在研究人的行为时，必然会涉及行为背后的潜在心理机制；另一方面，心理是一种内隐的活动，管理心理学在探索人的心理活动规律时，也需要通过观察分析人的外部行为来达到推断内部过程的目的。心理与行为的这种联系就决定了管理心理学与组织行为学之间的密切关系。

（2）研究内容上的联系

在研究内容上，管理心理学与组织行为学虽各有特色，但在总体框架上却无大的差别，研究的基本内容皆为组织管理活动中个体、群体、领导、组织等方面的心理与行为规律。因此两门学科在内容上十分相近，只是对同一问题的研究视角和出发点有所不同。

（3）研究目的上的联系

管理心理学与组织行为学的研究目的基本相同，即通过对组织管理活动中人的行为规律及其潜在心理机制的探索，揭示有关的规律，不断促进管理工作的科学化，持续提高工作绩效与管理效能，最终实现组织和人的全面发展。

二、管理心理学的研究逻辑

（一）两种取向的管理心理学

管理心理学是管理学和心理学交叉融合而形成的一门边缘学科，从事该学科研究的人员也主要来自于心理学和管理学领域。由于这两个领域的研究人员在学科背景、知识结构、看待问题的角度等方面存在差异，所以在这门学科不断发展的过程中，逐渐演变出了两种取向的管理心理学——心理学取向的管理心理学和管理学取向的管理心理学。

心理学取向的管理心理学是基于心理的或从个体角度出发的管理心理学。这种管理心理学看问题比较微观，注重个体层面的问题和行为背后的潜在机制，侧重于从心理学的角度出发，用心理学的理论、知识和方法来探讨相关问题，说明和解释相关现象。管理学取向的管理心理学是基于管理的或从组织角度出发的管理心理学，旨在为管理层或领导层出谋划策，教会他们如何管理员工，以便有好的业绩和绩效。这种管理心理学看问题比较宏观，也比较务实，侧重于用管理学的思想、原理和方法探讨相关问题，说明和解释相关现象。这两种取向已经在国内外不同版本的管理心理学著作中得到了体现。

两种取向的管理心理学在研究对象、内容体系等方面基本相同，不同之处在于研究问题的出发点不同。例如，在研究企业员工工作倦怠问题时，心理学取向的管理心理学更关注员工工作倦怠形成的心理机制，寻求导致其工作倦怠的心理因素；而管理学取向的管理心理学则更侧重从组织管理的角度出发，分析员工工作倦怠的形成原因，探查引起员工工作倦怠的社会、文化、组织条件等。

（二）管理活动中的人及其心理：手段与目的的辩证统一体

上述两种取向的管理心理学实际上触及到了一个非常重要而敏感的问题：管理活动的终极目的是什么？作为管理客体的人与作为管理主体的组织究竟是什么样的关系？人及其心理变化究竟是实现组织目标的手段或工具，还是组织管理活动的目的？换句话说，管理心理学研究组织情景中人的心理活动规律的目的何在呢？

管理学取向的管理心理学倾向于认为，探索管理活动中人的心理活动规律应该服务于管理的目的，是为了更加有效地、科学地开展管理工作，提高工作绩效，最终实现组织目标。按照这种观点，管理活动中的人及其心理变化只是实现组织目标的手段或工具，处于从属地位，因此研究管理活动中的人及其心理规律的管理心理学属于工具性或功利性的学科。而心理学取向的管理心理学则倾向于认为，管理活动中的人及其心理变化本身也应该成为管理的目的。管理心理学不仅应该服务于企业和组织目标，为该目标的实现提供依据、出谋划策，还应该服务于组织中的人，为企业和组织中人的全面发展提供依据和方法。当今的管理是以人性化管理或人本管理为主流的管理，在学术研究领域也已经出现了由人力资源管理（HRM）向人力资源发展（HRD）转变的趋势，人及其心理发展作为管理目的的地位更应该受到重视。

实际上，组织目标和人的发展目标并非一定是完全对立的关系，二者是互为因果、互为目的和手段的对立统一关系。揭示组织环境中人的心理和行为规律，一方面应该有利于

科学合理地开展管理工作，提高工作绩效，实现组织发展目标；另一方面应该有利于改善组织中人的心理状态，促进人的全面发展。组织目标的实现和组织中人的全面发展是互为目的和手段的，组织目标的实现为人的全面发展提供了条件，人的全面发展又为组织目标的实现奠定了基础。也就是说，管理活动中的人及其心理既是手段又是目的，同样组织也既是手段又是目的，二者都是手段和目的的辩证统一体。这种思想可以用图 1.2 加以更为详细的说明。

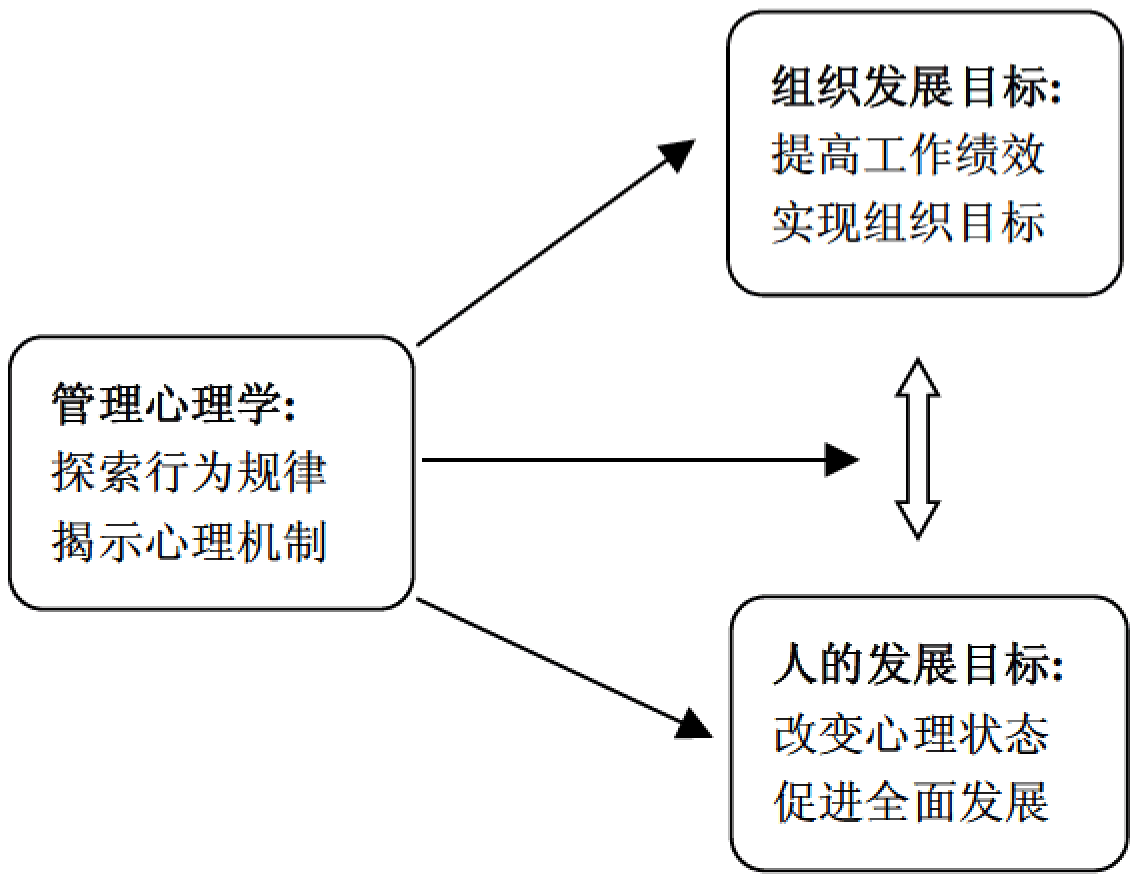


图 1.2　管理活动中的人与组织

首先，管理心理学具有双重目标或双重性。一方面，它要服务于组织发展目标，在这个意义上它具有工具性或功利性；另一方面，它要服务于人的发展目标，在这种意义上它具有人文性、教育性等非功利的特点。

其次，上述两种目标是交互作用、互相影响的。这种交互作用造成的四种状态如图 1.3 所示，其中 A 是理想状态，B 是不常见状态，C 是常规状态，D 是最糟状态。

第三，管理心理学除了直接服务于组织发展目标和人的发展目标之外，还需要对这两种目标之间交互作用的方式进行干预，协调二者之间的矛盾，调整二者之间的关系，使其达到相互促进、共同发展的状态，即达到图 1.3 中 A 所代表的理想化状态。

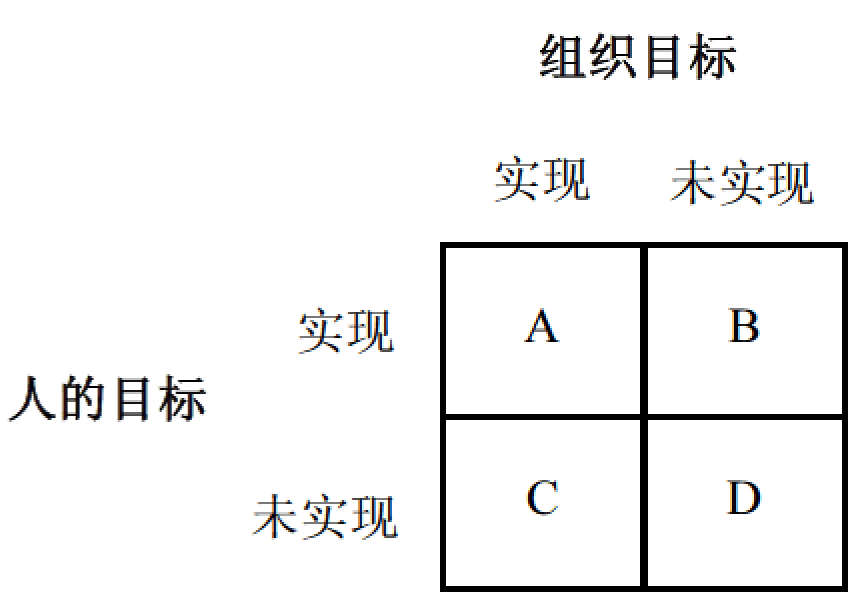


图 1.3　两种管理目标的交互作用产生的四种情况

（三）管理心理学需要思考的几个问题

如上所述，管理心理学是心理学的一个重要分支，属于应用心理学的范畴。因此，无论在理论研究层面上，还是在管理实践层面上，管理心理学都应该突出心理学的学科特色。那么，如何做到这一点呢？我们认为，至少应该解决好以下几个问题：

1．个体关怀或宏观关心

相对而言，管理学、组织行为学等学科在研究组织管理活动中人的心理与行为规律时，多是从较为宏观的层面上来考虑的。例如，管理学主要从组织层面上来研究，组织行为学主要从外显行为层面来研究等。作为管理学与心理学交叉而成的管理心理学，在探索人的心理与行为规律时应该表现出个体关怀的特征。所谓个体关怀，有两层含义：一是管理心理学应着重从微观层面来选择或理解问题，关心企业管理各个环节上与人有关的具体问题；二是管理心理学应以人为中心来思考和解决问题，对人身上存在的弱点和不足持较为宽容的态度，表现出人性化关怀的倾向。

具体而言，管理心理学的个体关怀特征应表现在两个方面：（1）在理论研究方面，管理心理学应以研究个体心理状态和特点为出发点，对体现心理学特色的课题，如员工激励、工作压力、工作倦怠、心理契约等，要在吸收借鉴心理学中成熟的研究工具和手段的基础上，不断进行深入研究，力图实现理论研究上的创新；（2）在实践应用方面，管理心理学在理解组织管理活动中涌现出来的各类实际问题时，应有独特的着眼点和分析框架，提出与管理学、组织行为学不同的解决方案，表现出个体关怀的特征。例如，在帮助企业设计薪酬方案时，管理心理学应关注这种方案与各阶层员工心理预期的匹配性，以及它对员工心理产生影响的长效性，而不是笼统地仅从组织行为学或管理学中分配公正这一角度来思考问题。

2．是什么或为什么

在研究组织管理问题时，管理学和组织行为学比较重视回答“是什么”和“怎么做”的问题。例如，在任何一本《组织行为学》教材中，都可以找到对企业普遍存在问题的描述和对相关应对措施的介绍，甚至还可以找到许多案例和练习性材料，但却缺少对相关原理和措施的深入心理学分析。这样做没有什么不对，但却是不够的。作为研究人的心理与行为规律的管理心理学，应该发挥心理科学得天独厚的优势（关注“为什么”的问题，能够提供实证资料），不仅要回答“是什么”和“怎么做”的问题，更要弥补组织行为学在回答“为什么”问题方面的不足。让人们既“知其然”，又“知其所以然”。例如，无领导的小组讨论、结构性访谈、评价中心法、360 度反馈等都是在企业人员招聘和员工考评方面比较流行的工具或手段。那么，这些工具或手段的理论依据是什么？为什么它们会比传统的方法好？这样做的效果如何？如何测定它们的效果？针对这些问题，管理心理学应该能够做出深层次的探讨，而不应仅仅谈论“是什么”、“怎么做”的问题。

3．实务或理论

通常在人力资源管理、管理学、管理心理学等学科的培训班上，学员们急切地想要掌握的是实务和操作方面的知识。这种心情可以理解，毕竟实务方面的知识针对性强，似乎也最有用。但实务和理论是不可分割的。没有深厚理论功底支持的实务训练只能是鹦鹉学

舌式的学习，这样获得的知识可迁移性和再生性差，任何人都不可能用在某个课堂上获得的实务技巧去应对管理实践中可能出现的各种局面。因此，我们不应该把管理心理学看成是一门纯实务的、技术性的学科，而应该认真掌握该学科中的各种管理学与心理学理论，唯有这样，才能更好地做好管理工作。也就是说，每一个管理工作者都不应该试图成为管人的“匠人”或“技师”，而应该试图成为管理的“大师”或“设计师”。从这个意义上说，系统地掌握管理学与心理学理论或许比掌握实务或案例更加重要。

4．行为或意识

实现对人的科学化管理是管理心理学的中心任务。一般说来，对人进行管理就意味着对人的改变。那么，究竟要改变人身上的哪些东西呢？多数人认为应该改变人的行为，而事实上我们更应该改变人的意识。行为是制度和规章的产物，而且具有高度的可模仿性和易获得性，你只要告诉员工应该做什么、怎样做，他就能够按照你的要求将有关的行为实现出来。然而，这样的行为是短效的、被动的，有时还会走样和变形，因为它没有员工意识的参与和支撑。举例来说，迎宾人员和出租车司机们都会按照公司的要求使用规范的礼貌性服务用语，但我们常常能够明显地感觉出他们的行为是机械的、不带感情色彩的，并不代表他们的意识。就连鹦鹉或机器人都能模仿出这样的行为。

只有干预和改变员工的意识，才能最终改变人的行为。而在改变员工意识或觉悟水平方面，管理心理学无疑扮演着重要的角色，因为管理心理学是研究组织管理活动中人的心理活动（意识）与行为规律的一门学科，它所提供的知识、原理和方法对于组织管理的柔性化、合理化、有效化、科学化都起着不可替代的作用。管理心理学工作者有责任和义务将管理心理学的方法和手段运用、推广到管理实践中，最终达到干预、提升员工和管理者意识水平的目标。

第三节　管理心理学的研究方法

任何学科的发展和进步都离不开方法的推动，正如苏联生理心理学家巴甫洛夫所说：“科学是随着方法学上获得的成就而不断前进的。研究方法每前进一步，我们便仿佛上升了一级阶梯，于是，我们就展开更广阔的眼界，看见从未见过的事物。”管理心理学的研究对象是人，人的心理和行为的复杂性决定了管理心理学的研究方法也是多种多样的。下面介绍几种常用的研究方法。

一、观察法

观察法是有目的、有计划地观察研究对象（被观察者）在一定条件下的言语、行为、表情等反应，从而分析其心理活动和行为规律的一种研究方法。观察可以以感官为工具，也可以利用录音、录像、摄影等现代技术设备作为辅助，来提高观察的效果。一般情况下，观察法可以按照以下两种维度进行分类。

第一，按照观察者所处的情境特点，可以把观察法分为自然观察与控制观察两种。自然观察是在完全自然真实的条件下观察他人的行为，而且被观察者一般不知道自己正在被

观察。例如，某销售主管若想弄清楚公司一名推销员绩效不佳的原因，就可以和该推销员一起销售产品，在工作过程中有意观察推销员的一言一行，从中发现问题的症结所在。控制观察是在限定条件下所进行的观察，也就是在操纵或控制一些条件情况下所进行的观察，被观察者知道自己处于被观察的状态。

第二，从观察者与被观察者的关系出发，可以把观察法划分为参与观察与非参与观察两类。参与观察是指观察者直接参与被观察者的活动，在共同活动中进行观察。相反，非参与观察是指观察者不参与被观察者的活动，以旁观者身份进行观察。

观察法的优点是目的明确，简易方便，所得资料比较系统真实。其缺点是，研究难以深入，所观察到的多为表面现象，取得的资料也较为肤浅，难以进行数量化的统计分析。鉴于此，在实际研究过程中，应把观察法与其他方法配合使用，以取得更佳的研究效果。

二、实验法

实验法是有目的地严格控制或创设一定条件，来引起某种心理活动或行为表现以进行研究的方法。根据实验场地的性质差异，可以把实验法分为实验室实验与现场实验两类。

实验室实验是在专门的实验室内，运用一定的仪器和设备严格地控制实验条件，以研究某种心理与行为现象的方法。例如，要考察表扬对人行为产生的影响，在控制其他无关变量的前提下，就可以设立表扬组（实验组），对被试良好的工作表现给予表扬；再设立一个对照组，不对该组被试良好的工作表现进行表扬。经过一段时间之后，比较两组后继工作的成绩，若表扬组优于对照组，就可以将其归结为表扬所致。实验室实验的优点是能够严格控制各种无关变量对实验结果的影响，研究结论具有可重复性，且所得数据具有精确性；缺点是人为性强，脱离实际生活情境。

现场实验是指在日常生活和工作的情况下，适当控制条件以研究某种心理与行为现象的方法。近年来，为了提高研究的外部效度，使研究结果更具普遍意义和可推广性，管理心理学家越来越重视现场实验研究。在管理心理学的研究中不乏现场实验的例子。例如梅奥（G.Mayo）在霍桑工厂进行的福利实验，就是通过提供或撤销某些福利措施，来探究其对生产总量的影响，这是典型的现场实验。现场实验的优点是能够结合日常生活和工作进行，避免了实验室研究的局限性，使研究结论更具备可推广性；缺点是不容易严格控制现场的无关变量，研究得出的结论可能会存在偏差。

三、问卷法

问卷法是研究者根据研究目的和任务，编制出内容明确、表达准确的问卷，让被试根据个人情况实事求是地做出回答，从而收集所需资料和数据的研究方法。问卷法是管理心理学研究常用的一种方法。常用的问卷形式有三种：是非式、选择式和等级排列式。

（1）是非式：采用只有“是”与“非”两种答案的问卷，让被试根据自己的情况对每个题目做出“是”与“非”的回答，不能模棱两可，也不能不回答。

（2）选择式：要求被试从并列的多种（两种以上）答案中按个人的实际情况选取一种或几种答案。

（3）等级排列式：在问卷中列出可供选择的多种答案，要求被试按其重要程度的次序

给以排列。

问卷法的优点是可以在较短时间内取得大量的材料，而且对调查结果采用统计方法进行处理分析，因此得出的研究结论更有普遍意义；缺点在于所得到的材料一般较难进行质化分析，不易把结论与被试的实际行为做比较。

四、访谈法

访谈法是研究者通过口头谈话的方式从被研究者那里收集第一手资料，了解被研究者心理与行为规律的一种研究方法。就研究者对访谈结构的控制程度而言，访谈可以分为三类：结构化访谈、无结构化访谈和半结构化访谈。

在结构化访谈中，研究者对访谈的走向和步骤起主导作用，按照自己事先设计好的、具有固定结构的统一问卷进行访谈。在这种访谈中，选择访谈对象的标准和方法、所提的问题、提问的顺序以及记录方式都已经标准化，研究者对所有的受访者都按照同样的程序问同样的问题。

与此相反，无结构化访谈没有固定的访谈问题，研究者鼓励受访者用自己的语言发表自己的看法。这种访谈的目的是了解受访者自身认为重要的问题、他们看待问题的角度及其表述方式等。无结构化访谈中，研究者只是起到一种辅助作用，尽量让受访者根据自己的思路自由联想。访谈的形式不拘一格，研究者可以根据当时的情况随机应变。

在半结构化访谈中，研究者对访谈结构具有一定的控制作用，但同时也允许受访者积极参与。通常，研究者事先备有一个粗线条的访谈提纲，根据自己的研究设计向受访者提出问题。但是，访谈提纲主要作为一种提示，研究者在提问的同时鼓励受访者提出自己的问题，并且根据访谈的具体情况对访谈的程序和内容进行灵活的调整。

访谈法的优点是简单易行，便于迅速取得第一手资料，因而使用范围较为广泛。其缺点是仅凭受访者的口头回答而做出的结论往往缺乏可靠性和真实性，因此，这种方法一般不单独使用，而应把它与其他研究方法结合起来运用。

五、测验法

测验法是指采用标准化的心理量表或精密的测验仪器来测量被试有关心理品质或行为的研究方法。量表是心理测验常用的研究工具，目前流行的测验量表种类繁多，大致有以下几种分类：按测验的内容可分为智力测验、个性测验、态度测验和能力测验等；按测验的方式可分为文字测验与非文字测验；按测验的方法可分为问卷测验、操作测验和投射测验三大类。

在管理心理学研究中，测验法常常作为人员测评的一种工具。例如，用智力量表测定组织成员的一般和特殊能力状况；用个性量表测定组织成员和领导者的性格特征等。测验法的最大优点是简便易行，测验内容广泛，具有较强的科学性，能够对研究的心理现象进行定量的分析。但测验法也存在一些问题，如心理测验的运用有一定难度，测验者必须经过专门的训练。另外，量表的设计、取样技术等都有较高要求，使用时若有不慎，就会使测验结果产生很大误差。

六、个案法

个案法是指对某一个体、群体或组织在较长时间里连续进行考察，了解、收集系统的资料，以便研究其心理与行为发展变化规律的方法。例如，研究者参与某企业的一个研发团队，通过长时间地体验生活，掌握了整个团队成员的心理与行为特点、团队绩效的状况、团队的人际关系等，并在此基础上进行深入分析，整理出能反映该团队特点的详细材料。

个案法的优点是呈现的内容丰富，有助于人们发现新问题，为研究者发现和提出新的理论假设奠定良好的基础。其缺点在于，这种研究一般都是描述性的，不容易在较短时间内做出有关因果关系的推论。此外，个案研究一般取样比较小，这就大大限制了研究结果的可应用性和普遍意义，而且得出的研究结论很难进行重复验证。

图 1.4 概括了上述几种研究方法。总之，它们都有其各自的应用价值，但也都存在一定的局限性。在具体研究过程中，究竟采用哪种方法较好，应视研究任务的要求和具体情境而定。通常情况下，管理心理学研究往往以某种方法为主，辅之以其他方法，使之取长补短、相得益彰，这样可以更准确、更客观地反映人的行为和心理活动的规律和特点。

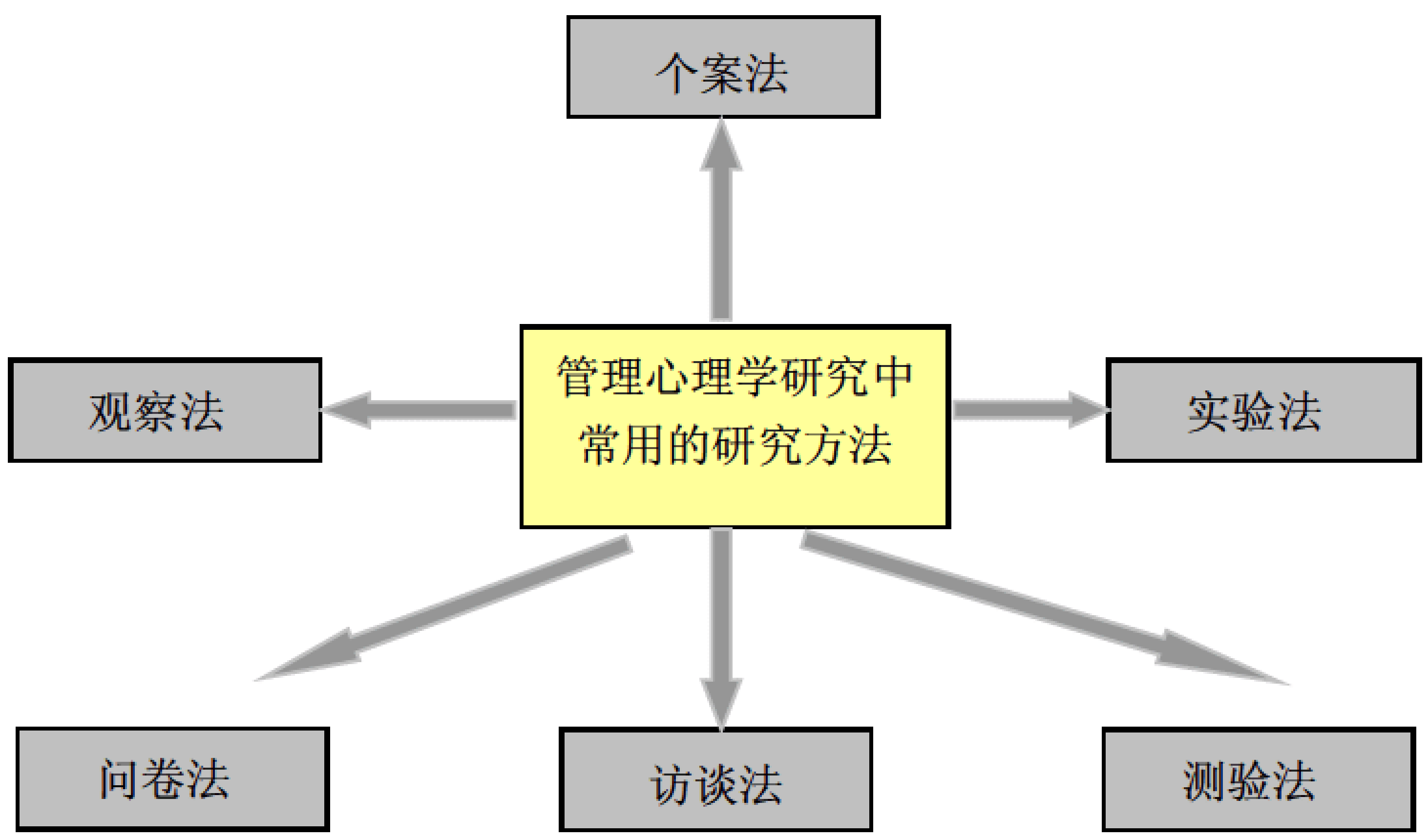


图 1.4　管理心理学的研究方法

本章小结：

1．管理心理学是研究组织管理活动中人的行为规律及其潜在的心理机制，并用科学的方法改进管理工作，不断提高工作效率与管理效能，最终实现组织目标与个人全面发展的一门学科。其研究内容涉及动机管理、认知管理、情绪管理、行为管理和组织管理等主题。

2．管理心理学是介于管理科学与心理科学之间的一门边缘性、交叉性学科，也是一门综合性和应用性很强的学科。它既是心理科学的一个分支，又是管理科学的一个重要组成部分。

3．管理心理学与组织行为学既有区别又有联系。区别主要表现为二者的研究侧重点不

同、理论基础不同、形成背景不同；联系主要表现为心理与行为的联系、研究内容上的联系、研究目的上的联系。有两种取向的管理心理学：心理学取向的管理心理学和管理学取向的管理心理学，二者在研究逻辑上有所不同。组织管理活动中的人及其心理发展既是手段又是目的，是手段和目的的辩证统一体。管理心理学要突出自身的学科特色需要思考以下四个问题：个体关怀或宏观关心，是什么或为什么，实务或理论，行为或意识。

4．管理心理学的研究方法主要有：观察法、实验法、问卷法、访谈法、测验法和个案法。

本章思考题：

1．什么是管理心理学？试比较本书的界定与其他同类著作界定的异同。
2．试分析管理心理学的研究内容。你赞同本书的观点吗？为什么？
3．试分析管理心理学与心理学、管理学、组织行为学等学科之间的关系。
4．试分析管理活动中的人及其心理发展的地位和作用。
5．试述组织发展目标与人的全面发展目标之间的关系。
6．管理心理学有哪些研究方法？试比较这些方法的优缺点。

第二章　管理心理学的历史沿革和现状

本章首先从管理心理学产生的时代背景和理论背景两个角度探讨其发展历史，然后对新时期该学科面临的机遇和挑战做一论述，最后探讨后现代思潮对管理心理学发展的影响。

第一节　管理心理学的产生

管理心理学在国外心理学界被称为组织心理学，是心理学领域的一个新兴的、重要的分支。管理心理学起源于20世纪初期，到20世纪60年代初期趋于成熟，成为一门独立的学科，其形成的主要标志是1958年美国斯坦福大学教授利维特（H.J.Leavitt）《管理心理学》一书的出版。从此管理心理学有了自己的名称和学科体系。

管理心理学的产生同当时的经济、社会发展及心理学和其他相关学科的发展有着密切的关系。

一、管理心理学产生的社会历史背景

自从有了人类，人们就开始了自己的管理实践，也开始形成管理心理学思想的萌芽。在中国，先秦时期是中国古代管理心理学思想产生和奠定的时期。在春秋战国大变革的时代，出现了“百家争鸣”的局面，诸子百家在政治、经济、军事、文化等方面提出了不同的管理主张。从人性与需要思想、用人心理、激励心理、领导心理、组织心理等不同的方面阐述了管理心理学问题。中国历代统治阶级的治国之道、用人之道（如人员甄选和测评思想）中也包含了丰富的管理心理学思想。在西方，古希腊、罗马、古埃及、巴比伦的历史记载中，到处可见管理心理学精辟的思想和巧妙的运用。当然，真正促进现代管理心理学产生的还是19世纪末至20世纪40年代前后的社会历史条件。从西方的社会历史发展情况看，19世纪末20世纪初，资本主义从自由竞争阶段向垄断阶段发展，生产的社会化程度不断提高。大企业的增多以及生产自动化程度的提高，特别是市场竞争的日趋激烈，迫使资本家考虑如何尽快改善劳动者的素质，缓和人机矛盾、劳资矛盾，提高工人的劳动积极性和工作效率，增强企业的竞争力。在解决问题的过程中，资本家发现改善企业内部的人际关系、领导行为有助于生产的发展。于是，他们就聘请大批心理学家从事这方面的研究。管理心理学应运而生。

两次世界大战的爆发，使心理学在军事管理领域得到高度重视和广泛应用，从而对管理心理学的产生起到了重要的推动作用。以美国为例，1917年4月6日对德宣战的同时，组织成立了17个战争心理问题研究委员会，分工研究新兵心理、服役人员心理、飞行员心

理、情绪不稳定者心理、士气与宣传训练心理、军纪心理、军队院校教育心理等数十项与军事有关的心理学问题。在第二次世界大战期间，美国训练了 1300 多名高级心理学工作者为战争服务，这项工作对美国取得战争胜利起到了巨大的作用。同时，军事上的协同、指挥、战斗效能的心理学研究也产生了丰富的管理心理学研究成果，推动了管理心理学研究队伍的壮大、研究方法的发展，扩大了管理心理学的影响，从而加快了管理心理学的发展。

二、管理心理学产生的理论背景

管理心理学的形成除了得益于 19 世纪末 20 世纪初整个社会经济的发展外，还与这一时期社会科学的发展有着密切关系。20 世纪初，泰勒（F. Taylor）倡导的科学管理运动和闵斯特伯格（H. Muensterberg）开创的工业心理学是管理心理学形成的先驱。真正推动管理心理学产生的是 1927 年由艾尔顿 • 梅奥（Elton Mayo）领导的霍桑实验。随后又相继出现了群体动力理论、需要层次理论、社会测量理论等，这些理论的形成与发展，为管理心理学奠定了比较充分的理论基础和学术背景。20 世纪 60 年代初期，管理心理学真正成为一门独立的学科分支而被人们广泛地接受。

（一）心理技术学

心理学知识在企业管理中的应用是从心理技术学开始的。心理技术学实际上是劳动心理学、工业心理学开始发展时的名称。心理技术学最早是由德国心理学家斯腾在 1903 年提出的，而最早具体对心理技术进行系统研究的是冯特的学生闵斯特伯格（H. Munsterberg）。闵斯特伯格受聘于哈佛大学，于 1912 年出版《心理学与工业效率》一书，提出心理学应对工人的适应能力与工作效率的提高做出贡献。他研究的问题涉及：如何用心理测验方法选拔工人，以适应工作的安排；如何改善工作条件，以减轻疲劳，使工人得到最大限度的满足和提高生产效率；如何创造心理条件，发挥一切有益于经济发展的心理氛围等。这些研究课题后来成为工业心理学、工效心理学、人事心理学、管理心理学的主要研究课题，正因为如此，闵斯特伯格被后人称为“工业心理学之父”。心理技术学的基本方向是解决人如何适应机器的问题，即研究人—机关系问题。闵斯特伯格的研究方向和路线，以及他所采取的方法，与管理心理学的发展方向是相一致的。但是，他所考虑的面比较狭窄，缺乏社会心理学与人类学的观点和论据，所以，他的工业心理学未能引起广泛的注意。后来由梅奥领导的霍桑实验增加了工业心理学的广度和深度。

（二）人群关系理论

人群关系理论的提出源于霍桑实验，其创始人为美国哈佛大学心理学家埃尔顿 • 梅奥（Elton Mayo）。实验在美国西方电器公司所属的霍桑工厂进行，为了提高工效，厂方组织专家在 1924—1932 年间开展了“生产效率与工作条件之间的关系”的系列实验研究。实验包括四个方面的内容：照明实验、福利实验、群体实验和谈话实验，分两个阶段进行：第一阶段是 1924 年 11 月—1927 年 4 月进行的照明实验和福利实验。这次实验的初衷是想证明工作条件与生产效率的关系是密切的，实验历时近两年半，结果令人迷惑不解，实验非

但没有证明工作条件与工作效率间的必然联系，反而显示了工作条件对提高工作效率仅有次要意义。带着对这一结果的困惑，1927 年 4 月梅奥召集了一批哈佛大学的教授，会同西方电器公司的人员组成了新的研究小组，再次到霍桑工厂进行第二次实验，即后人所说的谈话实验和群体实验。这次实验从 1927 年持续到 1932 年，长达 5 年之久。实验的最终结果表明，与工作效率相关程度最大的是工厂内自由宽容的群体气氛、工人的工作情绪、责任感等“软因素”，而不是诸如物理环境、物质刺激等“硬因素”。换言之，提高工作效率的决定因素是职工与管理人员间融洽的关系和企业内良好的人际氛围。

在霍桑实验的基础上，梅奥于 1933 年出版了著名的《工业文明中的人的问题》一书，首次提出了人群关系（或人际关系）理论。其理论要点包括：（1）工资不是企业员工的唯一追求，他们还有友情、安全感、归属感和受别人尊重等情感需求，管理者不能单纯从技术和物质条件着眼，而应首先从社会心理方面鼓励员工士气和生产积极性，以提高劳动生产率。（2）在正式群体中存在着非正式群体，管理者应充分重视非正式群体的作用，注意协调它们与正式群体的关系，以使管理者与员工、员工与员工能互相协作，充分发挥个人的作用，提高工作效率。（3）提出了提高员工满意度的新管理方式，认为管理者应当具有提高员工满意度的管理能力，应当采用以人为中心的管理方式；为了实现管理方式的转变，需要训练管理人员，使其掌握通过交谈了解人的思想感情的技能技巧，并提高协调正式组织与非正式组织的能力。可以看出，对管理心理学的产生而言，梅奥的人群关系理论起着直接的推动作用。梅奥首次将管理研究的重点从物的因素转到人的因素上来，将人际关系提升到管理工作的首位，不仅修正和补充了古典管理理论，而且开辟了管理研究的新方向，还为现代管理心理学的发展奠定了基础，对管理实践产生了深远的影响。

（三）群体动力理论

群体动力理论的创始人是德国心理学家勒温（Kurt Lewin）。勒温用物理学的力场概念和场论来阐释他的理论，因此，群体动力理论又称“场”理论。勒温把外界环境因素看成是外在的心理力场，把人未获得满足时的需要看作内部力场，人的行为动向取决于内部力场与环境力场的相互作用，但主要决定因素是内部力场。根据“场”理论，勒温提出了著名的行为公式：$B=f(P，E)$，其中，“B”表示行为，“P”表示个体，“E”表示环境。公式表明，个体行为是个体内部力场与所处环境力场的函数。后来，勒温把他对个体行为的观点应用于群体行为研究，提出了群体动力概念，意指群体构成要素相互作用的合力及其活动的方向。勒温认为，影响群体活动方向的因素是群体内部力场和环境力场的相互作用。勒温的群体动力理论对管理心理学的创立与发展有很大影响，是现代管理心理学理论体系中群体心理研究的重要依据之一。

（四）需要层次理论

美国心理学家马斯洛（A. Maslow）于 20 世纪 40 年代提出需要层次理论。他把人的需要从低到高分为五个层次，即生理需要、安全需要、归属与爱的需要、尊重需要和自我实现需要。马斯洛认为，要想最大限度地激发人的心理动力去努力工作，提高工作效率，就必须采取适当的管理措施去满足人的不同层次的需求。马斯洛的需要层次理论是管理心理

学激励原则形成的一个理论来源，对管理心理学的产生和发展都有极大的影响，直到今天，西方的管理心理学仍把这一理论作为其重要的理论基础。

（五）社会测量学

社会测量学的创始人是莫雷诺（J. L. Moreno）。1927 年莫雷诺迁居美国，从事社会心理学研究，提出了社会测量学理论。这种技术采用填写问卷的方法，让被试根据好恶感对伙伴进行选择，并将这种选择用图表表示出来，这样人们便可以对群体中各成员之间的关系进行分析。

尽管社会测量学有许多理论问题值得商榷，但是就其方法本身来说颇有实用价值。目前，在管理实践中人们仍广泛运用社会测量技术。

综上所述，正是特定的社会历史背景、众多的理论研究及人们对管理实践经验的总结与概括，使管理心理学最终在 20 世纪 60 年代登上了历史舞台。其后，管理心理学发展很快，且在管理科学领域逐渐成为一门重要的基础学科。

第二节　管理心理学的发展及现状

一、西方管理心理学的发展

在西方，管理理论的演变大致划分为三大阶段：古典管理理论阶段、行为管理理论阶段、现代管理理论阶段（包含数量管理理论、质量管理理论、权变管理理论、系统管理理论等）[①]，相应地，西方管理心理学理论的演进也大致划分为三个阶段。

（一）古典管理心理学阶段

大致形成于 20 世纪初—20 世纪 20 年代，代表人物是泰勒、法约尔、厄威克以及闵斯特伯格等。这一时期理论研究的特点是以工作效率为中心，研究的重点是如何通过改进工作条件、健全工作制度、用科学的管理方法来提高工作效率。当时人的重要性还没有凸显出来，人不受重视，盛行的人性观是经济人假设。

（二）行为管理心理学阶段

大致形成于 20 世纪 20 年代—40 年代，代表人物是马斯洛、梅奥等。受行为主义心理学的影响，这一时期理论的特点是以人的行为为中心，研究的重点是如何通过满足人的需求、调整人的行为、改善人际关系来激发人的创造性、主动性。管理研究的重点从物的因素转到人的因素上来，管理者的人性观开始转向社会人假设，当然，最终的落脚点仍是如何提高工作效率。

[①] 周三多．管理学．第 2 版．北京：高等教育出版社，2005

（三）现代管理心理学阶段

形成于 20 世纪 50 年代，至今仍处在发展过程中。这一时期理论研究的特点是比以往更强调从心理学和社会学的角度来研究管理，重视社会环境、个性发展及人际关系对提高工作效率的影响。它以现代管理理论为理论基础，以“人”为研究的中心。管理者的人性观经历了几次明显的变化：自我实现人——复杂人——文化人——决策人，对人的理解开始从多角度入手。这一阶段以来管理心理学得到了迅速发展，表现如下：

1．专业研究人员迅速增加，研究机构不断扩大

工业组织心理学会（Society for Industrial-Organizational Psychology，SIOP）是美国心理学会下属的专门机构之一。据该机构官方网站资料，1960 年美国工业与组织心理学会会员为 734 人，截止到 2000 年已增至 3601 人。而且进入这个领域的女性比例已经大幅提高。1989 年统计数字表明，得到工业组织心理学博士学位的人中大约有一半（46%）是女性，美国有 88 所大学培养管理心理学专业博士研究生，其中 46 所授予工业与组织心理学博士学位，42 所大学商学院授予组织行为学博士学位。

2．研究课题日益广泛深入

进入 20 世纪 90 年代以来, 组织变革已成为全球化经济竞争中管理心理学研究的首要问题，这方面研究主要探索组织变革的分析框架、理想的组织模式、干预理论以及变革代理人的角色。维克和奎恩（Karl. E. Weick & Robert E.Quinn,1999）研究发现，勒温传统的阶段性变革程序（即冻结——变革——再冻结）已逐渐被连续性变革程序（即冻结——再平衡——解冻）所取代，这种新的理论强调变革应是连续的、发展的、渐进的，尽管这些变革所进行的调整可能较小，但能够从根本上改变组织的结构和战略，保障变革的顺利实施，达到预期目的。

与组织变革密切相关的是领导行为研究。受权变理论的影响，近年来先后出现了多种领导理论，如通路—目标理论、领导—参与模式、生命周期理论。目前最有代表性的是费德勒（Fieldler）提出的认知资源利用理论，该理论强调：决定领导成效的关键与其说是领导者个人的智力和才能，不如说是使认知资源得以利用的条件。在组织变革中，管理决策因素显得尤为重要，因为组织结构调整总是在一定的风险情境下进行的。目前，在个体研究水平上，比较注重决策和判断中所采取的认知策略和判断决策问题；在组织研究水平上，主要分析不同背景下的决策模式、权利结构和参与机制，并特别重视决策技能的开发和利用。

激励问题是管理心理学研究的核心问题，过去曾产生了内容学派、过程学派和强化学派等有关激励的理论。目前，亚当斯（S. Adams）的公平理论对于薪酬设计的实际意义仍然受到普遍重视。此外，有关工作或组织承诺的研究日益受到关注，主要从工作价值观、职业发展、工作责任心、组织认同和对社会的态度等方面开展研究，探讨组织承诺对离职、工作满意感、工作安全感、人际关系的影响以及组织承诺的形成规律。组织文化也成为研究的热点，良好的组织文化有助于教育和引导员工形成好的认知、思考和感知问题方式，这方面的研究主要集中在组织文化的特点、结构和运行机制上。团队研究主要探讨团队的凝聚力、团队的构成、目标设定、团队内的关系、规范、角色、冲突和团队决策。管理的跨文化研究是适应跨国公司发展的新的研究方向，最有影响的是关于个人主义与集体主义

国民特性对组织管理影响的研究。

3．研究方法更加科学

为了提高研究结果的外部效度，当代的管理心理学研究大多采用现场试验、参与观察、大规模问卷调查等手段，开始从静态分析过渡到动态分析，从单一的心理学研究发展到心理学、管理学、社会学、人类学、政治学、经济学乃至数学的多学科综合研究。实验设计多采用多变量实验技术，并开始应用结构均衡方程、项目反应理论、关键事件法、计算机模拟等新方法、新技术。还有部分学者开始采用后现代的视角研究管理心理学问题。

二、我国管理心理学的发展

从整体上看，我国管理心理学的研究起步较晚。从 1978 年开始，随着改革开放的启动与逐渐深入，我国管理心理学者开始系统地引入国外管理心理学的理论和方法，逐渐完成了中国管理心理学的学科体系建设。从 1979 年中国心理学会筹建“工业心理学专业委员会”，1980 年 4 月该委员会正式成立算起，距今不过 20 余年的历史。在这短短的 20 多年时间里，管理心理学理论知识的普及和研究工作已取得不少的成绩，突出表现在以下三个方面。

（一）建立了许多学术组织与教学、研究机构

目前，我国有两个工业心理学的专门研究机构从事管理心理学研究：中国科学院心理研究所工业心理研究室、浙江大学心理学院工业心理研究所。20 世纪 80 年代以来，这两家单位已培养了管理心理学方面的数十名博士和百余名硕士，成为我国管理心理学研究的中坚力量。一些高等院校的心理学系、管理学院也都开展了管理心理学的教学与研究工作，如华东师范大学心理系、北京大学心理系等，为我国培养了不少管理心理学方面的人才。在研究资源方面，我国管理心理学界已具备从本科生到博士后的管理心理学后备队伍的培养体系；研究经费主要来源于国家自然科学基金委员会，部分来源于企业横向合作项目。2002 年 11 月，国家教育部高教司委托苏州大学举办了第一期全国管理心理学骨干教师培训班。很多高校都给学生开设了管理心理学方面的课程。目前，我国一些管理心理学研究者在国际应用心理联合会担任一定的领导职务，并参与了一些重要的管理心理学杂志的编辑工作，已在国际管理心理学界确立了自己的一席之地。

（二）开展了多方面的研究工作

我国的管理心理学研究者为数不多，研究的课题也未涉及管理心理学的所有领域，研究方法主要以吸收国外研究方法为主。首先，我国学者对工作动机、领导行为、管理决策、价值观、员工培训、人员选拔和组织变革方面，进行了较为系统的研究；其次，通过与国外工业与组织心理学家的合作研究，不仅缩短了与发达国家的差距，还丰富了国际管理心理学的知识体系；第三，近年来，管理心理学在国有企业改革、领导干部选拔、缓解员工工作压力、人才招聘和培训等应用领域也进行了一些研究和应用推广工作，发挥着越来越重要的作用。

（三）完成了一批研究成果

我国管理心理学的研究是从翻译和介绍国外的工业与组织心理学论著开始的。从 20 世纪 80 年代起，我国的学者陆续翻译了国外一些较有影响的管理心理学著作，如马斯洛的《动机与人格》、夏恩的《组织心理学》、麦考密克和伊尔根的《工业与组织心理学》等。我国学者自己编著出版的专著或教材也越来越多，如卢盛忠主编的《管理心理学》（1985）、苏东水著的《管理心理学》（1986）、周妙群编著的《管理心理学》（1990）、王重鸣编著的《管理心理学》（2000）、孔祥勇主编的《管理心理学》（2001）、俞文钊编著的《管理心理学》（2002）、朱永新主编的《管理心理学》（2002）、张向葵主编的《管理心理学》（2003）、薛振田编著的《管理心理学》（2005）、梁宝勇主编的《管理心理学》（2006）等。这些著作大多是为了满足高等院校和企事业单位的教学和培训需要而出版的，基本的体系差不多，但写作的角度和强调的重点各有不同。此外，还涌现了一批具有中国特色的管理心理学研究成果，如俞文钊关于同步激励论、公平差别阈的研究，凌文铨关于中国人内隐领导理论的研究，王重鸣关于管理决策相关问题的研究，时堪关于胜任特征和工作倦怠问题的研究等。

第三节　管理心理学面临的挑战和存在的问题

一、管理心理学面临的挑战

作为一门应用性很强的学科，管理心理学将随着时代的变迁和社会环境的变化而不断面临新的挑战。在当前情况下，管理心理学面临的挑战主要来自以下几个方面：

（一）经济全球化

经济全球化（globalization）使企业和其他组织面对的环境发生了根本的变化，企业组织的开放程度更高，竞争在全球范围内展开，必然会面临诸多新问题，过去的管理心理学理论和知识将不能适应新形势的要求，这就为管理心理学提出了挑战。就我国而言，加入 WTO 之后，加速了融入经济全球化浪潮的步伐，在由计划经济向市场经济转轨的过程中，企业将会在相当长一段时间内面对组织结构调整和发展带来的一系列管理心理学问题，如企业转制和重组、企业战略调整、管理决策、技术和管理创新等都会带来一系列心理学问题，亟待管理心理学理论予以解决。

（二）劳动力的改变——多样化

与过去的管理心理学相比，现代管理心理学面临的一个突出问题是劳动力的多样化。首先，现代社会职业分工越来越细，新的职业领域层出不穷，使得劳动力在工作性质、工作内容及在职业体系中所处的位置各不相同；其次，20 世纪 60 年代以来的妇女解放运动使得越来越多的女性加入到职业队伍中，也增加了劳动力的多样化色彩；第三，现代社会

是一个价值多元化的社会，不同国籍、不同年龄、不同性别的员工的价值取向和心理需求不同，即便是同一国籍、年龄和性别的员工也会由于家庭和教育背景及个性特征的不同而表现出不同的价值追求，特别是新生代的员工常常表现出与老一代的员工完全不同的价值取向，这在精神层面上助长了劳动力的多样化趋势。劳动力的多样化实际上意味着管理心理学研究的主体发生了变化，使得传统的管理心理学理论难以适应新形势的要求，从而为管理心理学提出了新的课题。

（三）新型的雇佣关系

现代社会组织和员工之间的雇佣关系也在悄然发生变化。在工作内容和时间上，现在的员工不像过去那样把工作的稳定性和安全性作为职业选择考虑的第一要素，而是注重职业提供的发展空间和机会。当一种工作不能满足自己的发展需要时，就会选择离开，因此人们的工作内容具有多变性，员工流动性增强，临时性工作增多，有人甚至干脆不加入任何组织，成为自由职业者。另一方面，在工作方式上，出现了虚拟团队、家庭办公等新的组织形式。这些也对管理心理学提出了新的挑战。

（四）信息技术与知识经济

现代社会是信息化的社会，计算机技术和互联网的迅速发展使组织的形式和组织内外的管理活动发生了翻天覆地的变化。首先，信息技术为较小的公司参与全球市场竞争提供了可能，使企业和组织面对的市场扩大了，也使知识和信息成为更为宝贵的财富和资源；其次，信息技术改变了组织的结构和形式，正在重新形成着组织，前文提到的虚拟组织、家庭办公等新的组织形式都是信息技术的直接后果；第三，信息技术改变了组织与组织之间联络和沟通的方式，使得企业的经营和商贸活动发生了变化；第四，信息技术既改变了组织内人们完成任务的方法，也改变了组织内人们沟通和一起工作的方式。这些变化所引起的人们的心理和行为问题亟需管理心理学加以研究和解决。

（五）团队化

现代组织越来越重视团队建设和团队合作，因此团队管理成为现代管理的主要形式。之所以如此，主要是因为现代企业的规模越来越大，组织的结构越来越复杂，团队能够比个人更加有效地解决复杂问题，能够减少组织层次，简化组织结构，还能够增强和培养团队成员的责任感和协作精神。由团队化带来的一系列问题也是管理心理学需要面对和解决的新问题。

（六）商业伦理的变化

随着社会的不断进步，人们的商业伦理观念在不断变化，伦理敏感度和道德紧张感也在不断提高。过去认为合乎商业规范的许多管理行为现在常常被视为违背了商业道德。例如，过去认为监控员工行为是天经地义的事情，现在有了窥视员工隐私之嫌；过去认为平均奖励、大锅饭是最公平的事情，现在被视为有失公正。此外，对员工形成刻板印象、使用同伴压力诱迫员工完成任务、采取组织政治手段达到管理目标等都被认为是有违商业道德的行为。这些变化也为新形势下的管理心理学提出了新的研究课题。

二、我国管理心理学存在的问题

（一）管理心理学理论的西方化

不可否认，我国管理心理学的学科体系源自于西方的管理心理学，只要将我国多数管理心理学教材的体系与 1958 年美国斯坦福大学教授利维特的《管理心理学》及目前比较权威的斯蒂芬 • P. 罗宾斯（Stephen P. Robbins）的《组织行为学》（第 10 版）的体系加以比较，就可发现这一点。此外，我国管理心理学的理论概念和研究内容也基本上沿用西方的概念体系。这固然有其客观的原因：一方面我国管理心理学研究的历史较短，基础比较薄弱；另一方面我国的管理实践相对落后。但管理心理学的理论、概念和方法具有明显的地域性和文化相对性，适用于西方的理论和概念未必适合我国的国情，所以有必要从我国的国情出发建立具有中国特色的管理心理学理论和概念体系。而且，这也是完全可能的，一方面中国传统文化中包含了丰富的管理心理学思想，为我们提供了充足的资料和理论起点；另一方面当代中国的改革开放，经济腾飞，尤其是企业组织的改制、转轨实践，为提炼具有中国特色的管理心理学理论提供了绝好的素材，我们完全应该有信心建立具有中国特色的管理心理学理论体系。

（二）管理心理学理论的学院化

管理心理学要想发展，离不开管理实践的推动，理论与实践的紧密结合是管理心理学发展完善的根本保证。一个很好的例子就是霍桑实验，在长达十年的实践研究基础上，梅奥提出了人际关系理论，发展了人性观，极大地推动了管理心理学的发展。但目前我国的现实状况是：一方面，从事理论研究的人员主要集中在高等院校和科研院所，与管理实践的分离与脱节使得许多研究过于“阳春白雪”，只管理论上自圆其说，却没有考虑到企业的实际状况和要求，或者建立的理论所要求的企业环境远远高于我国大多数企业的发展水平，从而导致很多理论无法在管理实践中应用推广；另一方面，企业组织中从事管理工作的人又常常由于理论素养不足，只能凭借经验开展常规事务性的工作，知其然而不知其所以然，一旦情况超出经验范围，就会手足无措，不知该如何应对。因此，我国的管理心理学必须走出“象牙塔”，从实践中来，到实践中去，从实践中汲取力量，为实践服务。

（三）管理心理学发展的滞后化

管理心理学作为一门重要的应用基础学科，发展的总体状况落后于管理实践要求。如图 2.1 所示，管理心理学属于微观理论层面的问题，人力资源管理是微观应用层面的问题；组织理论和组织发展则分别属于宏观理论和应用层面的问题。值得注意的是，当前人力资源管理的重要性已经凸现出来，组织发展的好坏快慢已经很大程度上取决于人力资源管理的水平和质量。企业的领导者也已经认识到了这一点，如在美国很多企业的人力资源总监在组织中的地位仅排在总经理、财务经理之后，在日本人力资源总监地位仅次于总经理。然而，管理心理学发展的滞后已经远远落后于人力资源管理实践发展的需要，严重地制约

了人力资源管理水平的提高。

	宏观	微观
理论	组织理论	管理心理学
应用	组织发展	人力资源管理

图 2.1　管理心理学与其他研究领域的关系

此外，管理心理学发展的滞后化还表现在：第一，由于国内的理论研究模仿多于创新，所以在对西方理论进行介绍时，语言转换和文献检索的时滞使得一些理论在西方风潮已过，而在我国却被冠以"新理论"、"新思想"而津津乐道。这种理论演进模式本身就使得我国管理心理学发展落后于西方数十年。第二，在研究内容上，我们总是喜欢对那些已经发生过的相当久远的问题进行回顾性的探讨，前瞻性和预测性的研究却相对匮乏。出现这种状况的原因一方面在于思想保守，担心前瞻性和预测性研究所包含的不确定性给自己带来学术风险；另一方面则源于我国管理心理学工作者自身知识结构和创新意识的欠缺。前瞻性研究要求研究者高瞻远瞩，对国家经济、政策、企业的未来走向等方面有深刻的洞察能力，这是以深厚的理论功底和学术素养为基础的，也需要求真务实的学术胆量和勇气，我国学者常常在此望而却步而造成了研究内容上的滞后。

（四）管理心理学理论的抽象化

具体表现在：第一，一直抽象地延用个体心理学、群体心理学、领导心理学、组织心理学四步曲的模式，将心理过程和管理过程分离开来。研究中理论成分多，实际探讨少。第二，抽象化地将政府管理和企业管理统称为管理，而实际上二者有巨大的差异。如何做到政企分开，有针对性、分门别类地探讨不同类型的问题尚未引起足够的注意。

除了上面讲的四个方面之外，我国管理心理学还存在以下问题：第一，研究力量比较分散，在针对国际最新趋势和国家社会经济转型要求方面，缺乏较为系统的研究规划；第二，国家自然科学基金委以及国家科技部虽然对于心理科学，特别是管理心理学在国家社会经济发展上有迫切的要求，但是经费投入较少，使得一些重要的、涉及管理心理学长远发展的重大课题得不到必要的支持，例如，管理科学部主要资助有直接应用价值的人力资源管理对策项目，而较少对管理心理学的理论前沿研究课题予以资助；第三，实验室设备急待改善。管理心理学是一个研究人—团体—机器系统的综合学科，随着高新技术的发展及其对于管理心理学研究的新要求，急待组建现代化的管理心理学国家开放实验室，以缩短与国外管理心理学的差距。目前，国外很多管理决策实验、情境模拟评价和训练、大样本调查和数据处理都离不开计算机网络的支持。应通过启动重大项目，组建适应跨世纪要求的管理心理学开放实验室。

三、管理心理学与后现代思潮

后现代（post-modern）思潮是对现代西方文化精神和价值取向的一次重大冲击，是对现代主流的科学认识观的怀疑，特别是对长期主导自然科学和社会科学的实证主义思想和原则的一种批判。它是 20 世纪中期西方发达国家由工业社会转入后工业社会（或信息社会）过程中出现的一种文化思潮，是在进入信息量剧增、知识爆炸、文化多元化的开放世界时，受作为主体的认知功能等因素的驱动而产生的人类认识观念的再反思。

后现代思潮的发展经历了两个阶段：早期阶段以激进和否定为特征，其主要流派有法国的德里达、福柯创建的后结构主义，德国的伽达墨尔的哲学解释学，以及以美国哲学家罗蒂等为代表的新实用主义；晚期阶段在回应和完善早期后现代主义的过程中，逐渐实现了以格里芬为代表的建构性后现代主义。

早期的后现代主义哲学家主张对结构的消解和对中心的摧毁。德里达认为，结构主义者滥用结构概念潜伏着某种危机，使用结构概念，意味着承认一种构架和一种占统治地位的原则。他主张消解整体性和等级性，即“解构”，以防止秩序的僵化和极权化。晚期的建构性后现代主义坚持对现代主义的批判，但是并不主张激进性的彻底摧毁。

后现代主义的主要观点可以概括为以下几点：

（1）后现代主义把人的认识观和科学观置于信息多样化和文化多元化的时代背景之下，强调认识主体内在概念系统对观察认知事物的先在作用，认为这具有不可忽视的合理性，符合辩证法。

（2）后现代主义不追求某种狭隘的唯一的方法，认为这可能误导人们的思想，而多元化的方法有益于人们在认识领域和科学研究中开放意识、展现思想。

（3）后现代主义强调科学与文化的联系，鼓励不同“文化成见”的沟通，认为这既有利于本土化研究，也有利于跨文化研究。

（4）后现代思潮强调方法的多元化，必然导致理论的多元化，认为这是契合当今多元世界发展的趋势的。

后现代思潮对管理心理学的影响可以从以下几方面来理解：

第一，后现代思潮已经引起并将会继续推动管理心理学研究主题、范式和方法的转变。[①]

（1）主题转变：主要表现在，在后现代思潮的影响下，许多研究者逐渐将注意力从过去聚焦于凝固、稳定的管理状态转向面对灵活的可变状态。当今流行的所谓变革管理、学习型组织、创新管理等概念和做法，就是这种转变的特殊表现。在过程管理方面，谈判、共识、双赢方式等受到了更多的重视；在人力资源方面，存在于主流之外的各种边缘人的生存状态、女性、人才多元化、民族和文化差异等也成为了管理心理学研究的重点；在组织架构上，关心组织存在的生态环境等也属于后现代管理研究的内容。在后现代思潮影响

[①] 吴岩．教育管理学基础．北京：清华大学出版社，2005

下，管理心理学研究的主题和内容正在沿着多元化的方向悄悄地发生变化。

（2）研究范式和学术规范的转变：改变追求统一范式的做法，强调个人的创造性和批判性，扭转少数学术精英对研究领域的控制局面，强调矛盾、讽刺、兼收并蓄和多元化因素等，是后现代学术规范兴起的标志。在文本表达方面，后现代主义研究者试图在论文中给读者提供尽可能多的生动案例，研究者和读者一起分享案例、证据、书信、照片、详细的对话记录及研究者亲眼观察到的事物。采用后现代方法的研究者非常重视写作过程本身，试图建立起一种友好、愉悦的陈述环境，与读者共同建构和分享基本的内容。

（3）方法转变：以往对管理现象的研究，无论是定性还是定量研究都习惯于进行大范围、大样本的调查研究和分析统计，这违背了后现代主义对世界的基本假设。由于大样本本身是不存在的，因此这些研究所得出的推论就显得没有实际意义。按照后现代主义者的主张，管理心理学研究应该从大范围的组织层面的研究转向对微观世界的独立而细致的案例性分析。此外，后现代主义还主张多种方法的兼收并蓄。

以新的态度对组织管理心理学的经典文献进行分析，也是后现代管理心理学研究的一种方法。波特（Burt，1987）曾经以全新的视角重新审视关于美国中西部地区内科医生滥用抗生素的经典著作，发现经典研究中存在着大量主观主义的内容，这些内容纯粹是为了进行学术对抗，有时作者还会为此改造数据以达到自己的目的。一些人认为，波特的贡献不仅仅在于对经典数据的权威解释提出了挑战，而且彻底改变了人们由于受社会影响坚守某些固定解释的信念。

后现代方法学的最大影响莫过于社会科学（包括心理学和管理心理学）领域质的研究的兴起。许多学术论文都采纳了这种方法。这种方法不仰仗于数据的统计分析，也不同于观察资料的报告，而是将重点聚焦在作者对问题本身的感受方面。在详细地研究过一个案例之后，作者不做大量的推广暗示，而是遵循这种分析方式，再度试图从这个分析中获得新的灵感。

第二，后现代思潮对管理心理学传统理论产生了冲击。

后现代思潮批判和解构的思想，必然对传统管理心理学的经典理论提出挑战和质疑。旧的理论被批判和解构，新的理论将会被创造。这种挑战和冲击，会导致整个管理心理学研究领域学术研究的混乱，研究者在没有权威的解构主义思想冲击下会出现方向的丧失。后现代思潮到底会冲击到什么程度尚没有定论，但是这种冲击是客观存在的。

第三，后现代思潮将会促进管理心理学向多元化的方向发展。

后现代思潮强调思想的多元化，这有利于管理心理学研究方向的多元化。多元化的研究方向必然导致管理心理学研究领域思想更加活跃，观点更加多样，从而呈现出百花齐放、百家争鸣的局面。

本章小结：

1．管理心理学的产生有其特定的社会和理论背景。19 世纪末 20 世纪初，资本主义从自由竞争阶段向垄断阶段发展，生产的社会化和自动化程度不断提高，催生了现代管理心

理学，同时期泰勒倡导的科学管理运动和闵斯特伯格开创的工业心理学是管理心理学的雏形或先驱，而真正推动管理心理学产生的是 1927 年由梅奥领导的霍桑实验。随后相继出现了群体动力理论、需要层次理论、社会测量理论等。20 世纪 60 年代初期管理心理学趋于成熟，成为一门独立的学科，其形成的主要标志是 1958 年美国斯坦福大学教授利维特《管理心理学》一书的出版，从此管理心理学有了自己的名称和学科体系。

2．西方管理心理学理论的发展大致经历古典管理心理学、行为管理心理学和现代管理心理学三个阶段。我国管理心理学的研究起步较晚，经历了从引进、介绍国外理论和方法到自主研究、发展的艰难探索过程，逐渐完成了我国管理心理学的学科体系建设。

3．新形势下管理心理学面临着来自经济全球化、劳动力多样化、新型雇佣关系、信息技术、团队化和商业伦理变化等方面的挑战。我国的管理心理学还存在着西方化、学院化、滞后化和抽象化的缺陷和不足，建立有中国特色的管理心理学理论和应用体系任重而道远。

4．后现代思潮拓展了管理心理学的研究视野和主题，克服了研究方法单一化等缺点，对现代管理心理学的发展将起到举足轻重的作用。

本章思考题：

1．试述管理心理学产生的历史和理论背景。
2．简述霍桑实验的内容和主要发现，分析其对管理心理学及当代管理实践的意义。
3．当代管理心理学面临哪些挑战？
4．我国管理心理学研究尚存在哪些问题？
5．后现代思潮对管理心理学都有哪些影响？

第三章　管理心理学的基础理论

本章主要探讨管理心理学的基础理论。众所周知，管理心理学是心理学和管理学交叉而成的边缘学科。管理学科的基础理论与心理学科的基础理论一起构成了管理心理学成长发展的理论基石。人性理论、博弈理论从管理学的角度为管理心理学奠定了理论基础，而精神分析心理学、行为主义心理学、人本主义心理学和认知心理学则从心理学的角度为管理心理学的发展铺平了道路。

第一节　管理学基础理论

管理心理学是把心理学，特别是社会心理学的理论在管理实践中加以应用的产物。人性问题的研究与管理心理学有着密切的关系，因为它直接涉及企业组织的管理者制定什么样的管理制度、采用什么样的管理方法、建立什么样的组织机构等问题。博弈理论是关于人们互动过程中策略选择的理论。这两种理论在员工管理、激励、绩效考核等方面影响和推动了管理心理学的发展，逐渐成为管理心理学的基础理论。

一、人性论

现代管理是以人为中心的管理，对人性、人的本质的观点不同，必然产生不同的管理原则和方法，因此人性问题就成为管理心理学研究的重要内容之一。

自古以来，人性问题一直是众多学科争论不休的问题。直到马克思主义诞生以后，才为正确理解这个问题指明了方向。马克思说："人的本质并不是单个人所固有的抽象物，在其现实性上，它是一切社会关系的总和。"这就是说，必须从人们在社会中所处的地位，从人们所处的社会关系来看待人性，不能抽象地看待人性问题。马克思主义关于人性、人的本质的理论是我们研究管理心理学问题、正确评价国外管理心理学中人性假设的基本指导思想，也是我们制定科学管理制度与管理方法的重要理论根据。这里的人性观，主要是指管理者对职工的需要和劳动态度的看法，并不涉及职工其他的观点和态度。

在西方管理心理学中，相继出现了四种主要的人性假设，即经济人假设、社会人假设、自我实现人假设和复杂人假设，近年来又出现了文化人假设和决策人假设。这些人性假设反映了西方管理学界对人性看法的发展演进历程。下面我们分别对这些人性假设的基本观点、所派生出来的管理建议及其局限性加以介绍和分析。

（一）经济人假设与 X 理论

"经济人"（Rational-Economic Man）也叫"理性经济人"，又称"实利人"。经济人假

设认为人的本性是追求自身利益的最大化，其工作动机就是为了获得经济报酬。

这种假设起源于享乐主义的哲学观点和亚当 • 斯密（Adam Smith，1723—1790）关于劳动交换的经济学理论，在 19 世纪末到 20 世纪初十分流行。

美国工业心理学家道格拉斯 • 麦格雷戈（Douglas M．McGregor，1906—1964）在《企业中的人性方面》一文中，把这种人性假设发展为 X 理论。X 理论的基本观点如下：

第一，多数人天生是懒惰的，都尽可能逃避工作；

第二，多数人都胸无大志，不愿负任何责任，甘愿受别人领导和指挥；

第三，多数人的个人目标与组织目标是相矛盾的，必须用强制惩罚的办法才能迫使他们为达到组织目标而工作；

第四，多数人干工作是为了满足基本的生理需要和安全需要，因此只有金钱和其他的物质利益才能激励他们努力工作；

第五，人大致可分为两类，大多数人具有上述特性，属于被管理者，少数人能克制自己感情冲动而成为管理者。

根据经济人假设，X 理论认为，管理过程中应该采取以下措施：

（1）采用任务管理的方式。管理工作的重点是提高生产效率，完成生产任务，至于人的感情以及道义上应负的责任是无关紧要的。管理就是计划、组织、经营、指导和监督。

（2）管理工作只是少数人的事，与广大工人无关，工人的主要任务是听从管理者的指挥。如泰勒曾指出，计划职能应和执行职能分开。

（3）实施明确的奖惩制度。在奖励制度方面，主要用金钱来刺激工人的生产积极性，同时对消极怠工者采用严厉的惩罚措施，即采用泰勒所鼓吹的“胡萝卜加大棒”政策。

在西方管理实践中曾经风靡一时的泰勒制就是经济人假设付诸实践的典型代表。

对于经济人假设及由它派生出来的 X 理论，应该从两个方面进行评价：

（1）局限性：首先，经济人假设和 X 理论是以享乐主义哲学为基础的。它把多数人看成天生懒惰、不喜欢工作的“自然人”。依据这种假设确定的管理理论、管理原则及措施，是以金钱激励为主的机械的管理模式，用权力严密地控制职工，因而不可能激发劳动者的主人翁精神和主动性、创造性。其次，经济人假设和 X 理论认为大多数人缺少雄心大志和责任心，只有少数人起统治作用，反对工人参与管理，这就把管理者与被管理者对立起来，不符合管理的本质。第三，经济人假设和 X 理论对工人的思想感情漠不关心，反对工人参与企业管理，甚至对于人不像牛那样愚蠢而感到遗憾，它不可能激发劳动者的工作动机，所以也不可能最大限度地发挥人的积极作用。

（2）合理性：经济人假设和 X 理论的产生有其特定的历史背景。在当时生产力不发达，物资比较匮乏的条件下，劳资矛盾突出的主要原因是工资待遇低下，使管理者考虑的重心落在了人性的自然层面。经济人假设的提出，对于缓和劳资矛盾，提高生产效率，提供了可操作的理论基础。即使是现在，尽管一些发达的资本主义国家一般都认为该理论已过时，但在一些欠发达国家或一些中小型企业的管理实践中，仍然能看到这种理论的影子。

（二）社会人假设和人群关系理论

“社会人”（Social Man，有时也译为“社交人”）假设产生于 20 世纪 30 年代至 50 年

代。社会人假设认为，人的最大动机是社会需求，只有满足了人的社会需求，才能对人有最大的激励作用。良好的人际关系是调动人的生产积极性的决定性因素，人们在工作中得到的物质利益对于调动生产积极性只有次要意义。这种假设是建立在社会心理学家梅奥（Elton Mayo）教授提出的人际关系学说的理论基础上的。梅奥在著名的霍桑试验以后，提出了人际关系学说，在1933年出版的《工业文明的人性问题》一书中，提出社会人假设的基本观点：

（1）人是社会人。不能把人看成是单纯的经济人，金钱不是激励职工积极工作的唯一动力，员工的社会心理需求才是影响职工积极性的主要因素。

（2）"以人际关系为中心"的管理。认为生产效率的高低主要不决定于工作方法和工作条件，而取决于职工的"士气"。而士气取决于职工在家庭、企业及社会生活中的人际关系是否协调一致。

（3）重视"非正式群体"。霍桑试验揭示了正式组织中存在着"非正式群体"。这种无形的组织有其特定的规范，影响群体成员的行为，应重视并加以研究。

（4）建立新型领导方式的必要性。领导者要善于倾听职工的意见，在研究工人们合乎理性行为的同时，还必须了解工人们不合乎理性的行为，如情绪、感情及人的潜意识等引起的行为。要设法使正式组织的目标同非正式组织的社会性需要相协调，使工人愿意为达到组织目标而协作和贡献力量。

基于"社会人"假设，梅奥提出了"人际关系"理论，并认为管理者应该在这几方面发挥作用：

第一，管理人员不应只注意生产任务的完成，而应该注意工人的各种需求，把重点放在关心人、满足人的需要上；

第二，管理人员除了指挥、监督、计划和组织，还要关注职工之间的人际关系，努力提高职工的认同感、归属感、整体感，激励职工对组织的奉献精神，培养职工的群体意识，增进组织的凝聚力；

第三，在实施奖励时，主张实行集体奖励制度而不主张实行个人奖励制度；

第四，管理人员的职能也应有所转变，他们不应只限于制定计划、组织工序、检验产品，而应在职工与上级之间起联络人的作用，担负起上下级之间信息沟通的责任。一方面，要倾听职工的意见，了解职工思想感情和需求；另一方面，要将上级的意图和部署向下级传达和贯彻；

最后，提出了"参与管理"的新型管理方式。所谓"参与管理"，就是在不同程度上让职工或下级参与企业政策的研究与讨论。实验研究和管理实践都证明了参与管理比传统管理更加有效。

目前，日本很多企业推行"参与管理"制度。具体方法有，成立工人俱乐部、派送生日礼物、与工人一起搞社交活动、郊游聚餐、成立健康管理室等，这在一定程度上缓解了劳资矛盾，对生产效率的提高有一定的促进作用。

社会人假设及人际关系理论的局限性在于尽管突出了人际关系对调动员工积极性的重要性，却忽略了工作任务、工作本身对人的重要作用。其合理性表现在：（1）加深了对人

性的认识。“社会人”假设承认人是社会实体，认为在看到人的自然性需要的同时，更要重视人的社会需要。（2）丰富了激励理论。从满足人的社会需要入手进行管理，可以收到比单纯的物质刺激更好的效果。（3）改变了管理重心。开创了以人为本的管理模式，促进了管理水平和生产效率的提高。

总之，从经济人假设到社会人假设是管理思想的巨大进步。按照马克思主义的观点，人的本质属性是人的社会性，而社会人假设接近这样的看法。

（三）自我实现人假设与 Y 理论

自我实现人（Self-Actualizing Man）也叫“自动人”。这种人性假设认为，人们力求最大限度地发挥自己的潜能，表现出自己的才华，只有这样，才会获得最大的满足感。

自我实现人假设产生于上世纪 50 年代，是由美国著名心理学家马斯洛（Abraham H. Maslow）、阿吉里斯（C. Argyris）和麦格雷戈（Douglas McGregor）等人提出来的。

麦格雷戈总结并归纳了马斯洛、阿吉里斯以及其他人的类似观点，并结合管理问题，提出了与 X 理论相对立的 Y 理论，其主要观点是：

第一，一般人都是勤奋的，如果环境条件有利，工作如同游戏或休息一样自然。

第二，限制和惩罚不是实现组织目标的唯一方法。人们在执行任务中能够自我指导和自我控制；

第三，一般人在适当条件下，不但能够接受责任，而且会追求责任。逃避责任、缺乏抱负以及强调安全感，不是人的本性；

第四，在人群中广泛存在着高度的想象力、智谋和解决组织中各种问题的创造性。

最后，在现代工业化社会条件下，普通人的智力只得到了部分的发挥。

麦格雷戈提倡 Y 理论，反对 X 理论。他的 Y 理论实际上是自我实现人假设的具体化。与 Y 理论相对应，管理措施应发生如下改变：

（1）管理重点的改变。管理重点不应只放在计划、组织、指导、监督和控制上，而应更加重视人的因素，更加注意人的价值和尊严，尽量把工作安排得富有意义、具有挑战性，使职工通过工作过程和工作业绩得到满足、发展。这样既克服了经济人假设把管理重点放在任务上而轻视人的片面性，又克服了社会人假设仅重视人的因素与人际关系而忽视工作任务本身的局限性。自我实现人假设主张创造适宜的工作环境和工作条件，充分发挥人的潜力和才能，发掘个人的特长和创造力。

（2）激励方式的改变。经济人假设靠物质报酬激励职工的积极性，社会人假设靠良好的人际关系来调动人的积极性。自我实现人假设认为，对人来说，最根本的、长远起作用的是内在的激励因素，即在工作中获得知识，增长才干，充分发挥自己的潜力等。只有内在奖励才能满足人的自尊和自我实现的需要，从而极大地调动职工的积极性。

（3）管理制度的改变。自我实现人假设主张下放管理权限，建立决策参与制度、提案制度、劳资会议制度，以及制定发展计划，让职工自己选择工作。

（4）管理者职能的变化。从自我实现人假设出发，管理者的主要任务在于尽可能为人们充分发挥自己的聪明才智创造适宜条件，减少和消除职工在自我实现过程中所遇到的障碍。

自我实现人假设与 Y 理论的局限性在于，从理论上看，仅仅把管理建立在人们自我实现的潜能和勤奋向善的本性基础之上是靠不住的。而且，人性的发展和人的自我实现并非是自然成熟的过程，而是社会现实和主体能动性综合作用的结果。其合理性表现在，强调为职工创造学习与深造的机会和条件，以充分发挥其聪明才智，主张给职工以“挑战性”的工作，让他们在自己工作的成就中得到乐趣和满足，提倡建立决策参与制度、提案制度等，这些对于管理实践都有重要的参考价值。

（四）复杂人假设与超 Y 理论

复杂人（Complex Man）假设是 20 世纪 60 年代末至 70 年代初由组织心理学家雪恩（Schein）等人提出来的。复杂人假设认为，人不只是单纯的经济人，也不是完全的社会人，更不会是纯粹的自我实现人，而应该是因时因地、因各种情况采取适当反应的复杂人。

不仅人与人之间的个性不同，而且同一个人，在不同年龄、不同时期、不同地点也会有不同的表现。人的需要和潜力会随着年龄的增长、知识的增加、地位的改变而有所变化，不能用单一模式去生搬硬套。复杂人假设就是以这样的观点为基础，力求合理说明人的需要与工作动机的变化规律。依据复杂人假设，1970 年莫尔斯和洛西（J. Malse & J. W. Lorsch）提出了一种新的管理理论，称之为权变理论（Contingent Theory），认为人们怀着不同的需求加入到组织中来，由于需求的多样性，以 X 理论为指导的管理方式和以 Y 理论为指导的管理方式都有其适用的环境，因此对于不同的企业或不同的人应采取不同的管理方式。

这一理论也叫超 Y 理论，其主要内容如下：

（1）人的需要是多种多样的，并且随着人的发展和生活条件的变化而变化。每个人的需要各不相同，需要层次也因人而异。

（2）人在同一时间内有各种需要和动机，它们会发生相互作用，并结合为统一整体，形成错综复杂的动机模式。

（3）动机模式的形成是内部需要和外界环境相互作用的结果，人在组织环境中，工作与生活条件不断变化会产生新的需要与动机。

（4）一个人在不同单位工作或同一单位的不同部门工作，会产生不同的需要。

（5）由于人们的需要不同、能力各异，对于不同的管理方式会有不同的反应，因此没有一套适合于任何时代、任何组织和任何个人的普遍行之有效的管理方法。

根据复杂人假设提出的权变理论，并不是要求管理人员采取完全不同于前述三种假设的管理措施，而是要求管理者根据具体人、具体工作性质和环境，灵活地采取不同的管理措施。其主要的管理措施有：

（1）管理者要有权变论的观点。领导方式要随实际情景而变，若企业任务不明、工作混乱，应采取较严格的领导方式，以建立良好的工作秩序；若企业任务明确，分工清楚，工作秩序井然，则应采用民主的、授权的领导方式，以充分发挥下属的积极性和主动性。

（2）管理者的管理策略与措施不能过于简单和一般化，应该根据具体情况具体分析，采取灵活多变的管理方法。

（3）采取不同的组织形式提高管理效率。根据工作性质的不同，采取灵活多变的组织形式。

（4）注重个体的差异性。善于发现职工在需要、动机、能力、个性等方面的个别差异，因人、因时、因事、因地采取奖惩措施。

复杂人假设与超 Y 理论的局限性在于，只强调人们之间差异性的一面，而忽视了人们共性的一面，不利于发现和寻找管理的一般规律。其合理性在于包含了辩证法的因素，从人们之间的差异及其与环境的关系的角度出发，强调针对不同的具体情况和不同的人采取灵活的管理方式，这对于管理工作无疑有一定的启发意义。

上面介绍的四种人性假设与不同层次需要的关系及其管理意义可以用图 3.1 来加以说明。

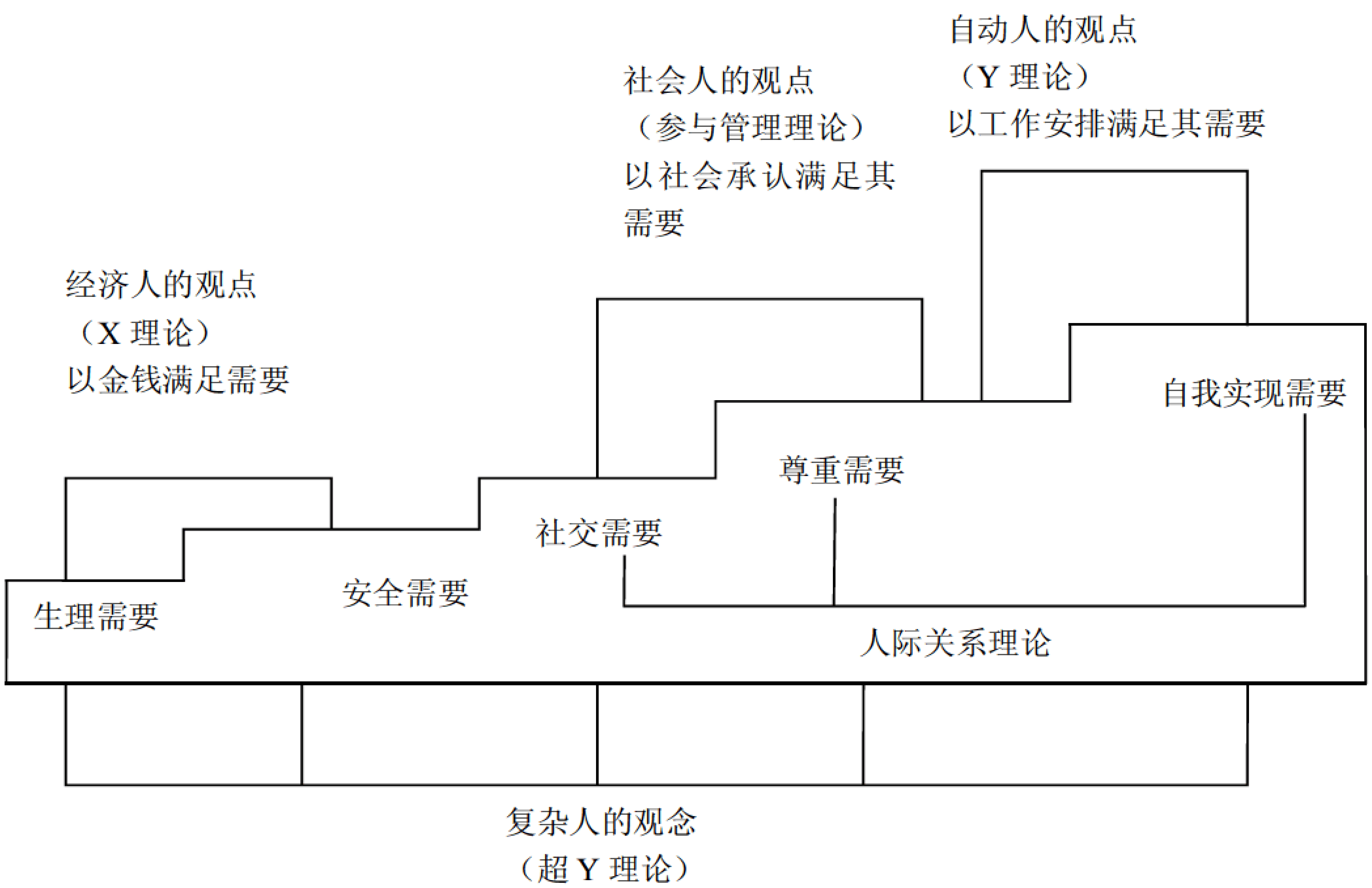


图 3.1 人性假设、需要理论与管理理论的比较

（五）文化人假设与 Z 理论

文化人假设是 20 世纪 80 年代初西方关于企业文化问题研究浪潮的结晶。文化人假设认为，人是文化的产物，人的心理与行为归根结底由人的价值观等内在因素决定。美国加州大学日裔管理学家威廉·大内 1981 年发表了《Z 理论——美国企业界怎样迎接日本的挑战》一书，成为企业文化理论的代表。

文化人假设的主要观点可以概括为以下几个方面：

（1）人是文化的产物，即使是人的一些生物性行为也打上了某种文化的烙印。

（2）同一文化背景下的人具有共同的行为模式，如不同民族的人各有自己的民族性格或国民性，不同地区的人也有不同的地域性格，这其实是人类行为文化模式的不同造成的。

（3）人格的塑造最核心的是对人的价值观、行为方式的培养和训练。

（4）群体或企业行为的效率主要依赖于群体成员是否具有共同的价值观和行为准则。

文化人假设的着眼点在于管理模式的建构，而管理模式的形成与培育职工价值观和树立企业形象这两方面的工作密不可分。威廉·大内认为，二战后日本企业发展速度之所以远远超过美国，原因在于日本企业的管理模式不同于美国企业。美国企业要提高效率以迎接日本的挑战，需要把J型管理模式按照美国的方式进行修改，以适应美国国情。威廉·大内把经过修改的理想的美国管理模式称之为Z理论。

Z 理论的主要特点是强调以下几点：（1）长期雇佣制；（2）集体决策；（3）个人负责制；（4）缓慢的评定和提升；（5）适度专业化的职业道路；（6）含蓄控制和明确控制相结合；（7）整体关心，包括对职工家庭的关心。

文化人假设和Z理论提出了新的管理视角，认识到了企业文化对企业的重要作用，强调企业发展离不开企业员工的价值观、企业精神、企业形象的培育和塑造。在此基础上着眼在于管理模式的建构，提出了新的管理模式，从而使管理者透过企业管理的理性层面挖掘出内在的、无法用理性加以涵盖的文化层面，即职工的文化素养。

（六）决策人假设

决策人假设是巴纳德率先在管理理论中提出来的，而后由心理学家西蒙加以完善而成。前面探讨的理论无一例外地以调动劳动者的工作积极性为目的，这固然是提高管理绩效的前提条件，但这并不能保证管理绩效的必然提高，因为尽管在工作任务简单、经济关系单一的条件下，单靠调动劳动积极性可以获得良好管理绩效，但在环境（特别是市场环境）对组织的影响日益强大，以及组织对个人行为的调控更加多样的条件下，仅凭调动工作积极性来提高管理绩效的做法是远远不够的，只有充分调动工作者的“大脑”，才能获得良好的管理效果。实际上，20世纪40年代以来，社会生活中发生了许多引人注目的重大变化。首先，市场经济在其发育过程中迅速超越行业、地域乃至国家界限，出现了全球化趋势；其次，市场经济运作中的不尽人意使凯恩斯主义登台亮相，组织面对的生存环境，不仅有经济上的竞争对手，而且有各类行政性组织及相应法规；再次，随着生产技术的进步，组织内部的技术结构日益复杂，仅靠规章约束、经济奖惩、诱因激励等手段来规范人们的行为和工作方式，已很难收到良好的效果。正是在此背景下，决策人假设应运而生。

决策人假设包括如下要点：

（1）每个人都是自主决策的行为主体，而决策本身并非不可分解的基本单位，而应视为由前提推出结论的过程，决策前提包括价值要素和事实要素。

（2）决策前提的引入既与决策者本身的素质有关，也与决策者所处的环境有关。

（3）组织并不代替个人做决策，但是组织可以通过提供相关的事实前提和价值前提以影响个人决策。

决策人假设的局限性在于过分强调组织应尽量、尽快将环境影响、决策前提的变化程序化、规范化，强调组织的稳定、有序，这就在事实上忽略了个人在工作中的创造性。个人的工作能动性是建立在被动接受信息基础上的，因而员工主体能动性的发挥受到了一定限制。

其合理性表现在：

第一，决策人假设把人的行为放在特定的组织背景下，不对人的活动目的及相应手段作永恒不变的先验设定，而把目的和手段看成是可在一定范围内加以调节的变量。

第二，决策人假设的着眼点不是单个人的工作积极性，而是群体合理决策中的行为协调，强调了激发工作能动性的重要意义和可能途径。一方面，它提示组织要充分关注组织的生存环境，并努力寻找适应环境的一种管理模式，使组织决策与个人决策相协调，进而激发员工的工作能动性；另一方面，它提示组织要充分关注自身所拥有的信息条件，在采集、存储、加工、使用信息方面为劳动者发挥工作能动性提供帮助。较之仅关注劳动积极性的经济人和社会人而言，它强调了创造条件（特别是信息条件）以激发工作能动性。

二、博弈论

（一）博弈论简介

博弈论（Game Theory）被称为20世纪人类最杰出的贡献之一。1994年，博弈论专家纳什（Nash）获得诺贝尔经济学奖。在西方社会科学中，博弈论被评为“纪念西方文明发展的十八座里程碑”奖章的第十七位荣膺者，这不仅是对其理论的肯定，而且显示了其蓬勃发展的趋势。目前博弈论已成为众多领域的基本分析方法。

一般认为，博弈论萌芽于20世纪20年代初，法国数学家波莱尔最先提出了“策略”和“混合策略”概念，并用数学方法处理了一些简单的博弈问题。博弈论的诞生则开始于冯•诺伊曼（Von Neumann）1928年对博弈论基本定理——极小极大值定理——的证明。博弈论创立的标志是冯•诺伊曼和奥•摩根斯坦（Morgenstein,O.）1944年合著的《博弈论与经济行为》一书。他们的贡献主要是创立了博弈论研究的基本概念，提出了二人零和博弈的完全解决以及对合作博弈的深入研究。现在应用更为普遍的非合作博弈理论的创立则是以纳什1950年的博士论文《非合作博弈》为标志，该文的主要贡献是提出纳什均衡的概念。此后（20世纪70年代），美国的海萨尼和德国的塞尔顿的不完全信息博弈理论工作进一步完善了非合作博弈理论。博弈论在我国的研究开始于20世纪90年代，大家公认的国内关于博弈论研究的起始标志是1996年张维迎教授《博弈论与信息经济学》一书的问世，从此博弈论在经济、管理、政治、军事等各个领域得到了广泛应用。

博弈论是研究人们互动行为决策的理论。这一理论为思考人类行为提供了一个很好的方式：每个人采取特定行动能够得到什么好处，不仅取决于这个人自己的行动，也取决于其他人对这个人特定行动的反应，即在一个交互式的行动环境中，人们的利益是相互依存于对方的行动的。因此，每个人在追求自己利益而采取自己的行动时，必须考虑到其他人的行动选择。

博弈论本身包含三个最基本的要素：（1）局中人（player），即参与博弈的主体，又叫参与人；（2）策略（strategy），即各个局中人可以选择的战略行动空间；（3）支付（pay－offs），即各个局中人在从博弈的所有局中人的战略组合状态下能够得到（或失去）什么。

博弈均衡是一种状态，在这种状态下，没有任何一个参与人能够通过偏离这种状态获得更大的好处。博弈均衡的一个基础概念是纳什均衡，在纳什均衡基础上还发展出了许多均衡概念；但是，博弈均衡所指的这种状态，在任何博弈中都是可以成立的概念。目前，经常出现在管理学中的博弈概念及策略主要有“囚徒困境”与价格策略、“智猪”策略与中小企业搭便车、合作博弈与战略联盟等。

（二）博弈论在管理心理学的应用

1．理性人的悖论思考与决策的满意原则

传统经济学的理性经济人假设认为，人都具有完全的理性，并以此来实现利润最大化。完全理性假设适合以下三种条件：第一，可供选择的备选行为是已知且固定的；第二，各种选择的结果的概率分布是已知的（对主观概率而言）；第三，目的是为了使一个给定的效用函数的期望值最大化。这种假设条件非常严格，与实际生活不符，所以在运用方面受到很大限制。

1978 年诺贝尔经济学奖得主西蒙（H. Simon）认为，完全理性是不存在的，他以“有限理性”和“满意性原则”取代“完全理性”和“最优化原则”，进而解释人类的行为（详细内容将在第 9 章中介绍）。

2．企业对员工激励与约束的博弈

我们可以在博弈论的框架下来分析管理活动中激励和约束机制的设计。博弈论认为，“任何一种有效的制度安排必须满足激励相容（incentive compatible）或自选择（self-selection）条件”。这就是说，决策者必须在考虑其他局中人反应的基础上选择自己最理想的行动方案。

激励与约束机制的设计是博弈论和非对称信息博弈论在管理心理学领域的运用和发展。组织的任务是如何通过合理配置组织资源，激发、引导和约束组织成员的行为，实现管理制度化和个性化的平衡，达成组织和个人目标的一致。管理者对被管理者实施激励和约束时，必须考虑被管理者的需求及可能采用的反应对策，必须在充分满足被管理者效用最大化的前提下去实现组织效用的最大化。博弈论提供的框架具有分析连续、多阶段互动关系的能力，因此只有遵循博弈规则研究管理激励与约束机制，才能使所设计的激励和约束机制符合管理实践的需要。组织目标具有规范约束效力，这也是激励约束的立意基础，所以博弈过程是在目标约束下的博弈。此外，在企业与员工之间建立激励博弈的机制时，还应强调公平是留住员工、稳定队伍、创造价值的关键所在。

3．绩效考核博弈

在绩效考核博弈中，博弈双方为参与考核的主管与员工，博弈对象为员工的工作绩效，博弈方收益为考核结果。

博弈期间，员工的合作策略指员工愿意根据实际工作绩效做出客观的评估，而员工的不合作策略在实际中大多表现为有意识地掩盖自己的错误或拔高自己的工作绩效。类似地，主管的合作策略指能够根据员工的实际工作绩效做出客观的评估，而主管的不合作策略指随意做出考核结果，更多表现为采取宽容下属的决策。当员工及主管同时采取合作策略时，则人力资源部可以得到较为公正客观的数据和精确的考核结果，从而做出较为适当的处理

结果，即与员工的工作绩效能有效结合；当员工采取合作策略而主管采取不合作策略时，人力资源部得到的数据则过多倾向于以员工提供的材料为主，从而使考核结果有利于员工。同时人力资源部会得出主管未能完成他的一部分职责的结论，因此影响到主管的工作绩效评估，从而影响到主管的晋升及加薪；当员工采取不合作策略，而主管采取合作决策时，则处理结果中主管所占的比重有较大程度的提高。人力资源部认为员工缺乏应有的敬业精神，从而影响到员工的长期发展机会，而主管则得到认可，从而为他进一步的发展提供良好的基础。如果员工与主管均采取不合作策略，人力资源部缺乏客观的资料来源，从而对员工和主管的绩效评估结果均缺乏依据。随着考核博弈的不断重复，员工与主管都希望自己的策略收益最大化，他们会发现合作策略是最有可能实现此目标的策略。在实际工作中，由于主管在考核中通常占有较高的比重，所以主管的个人倾向往往对考核结果有较强的影响力，而且考核常常是无限次的重复博弈，因此员工为了追求效用最大化，很有可能根据主管的个性倾向调整自己的策略。

第二节　心理学基础理论

管理心理学的发展受到了不同研究取向的心理学理论的影响，其中尤其以精神分析心理学、行为主义心理学、人本主义心理学、认知心理学为重，它们为管理心理学学科大厦的建构做出了各自独特的贡献，共同构成了管理心理学的基础理论。

一、精神分析心理学

美国著名心理学史家波林（E．G．Boring）在其巨著《实验心理学史》一书中曾这样写道：“谁想在今后300年内写出一本心理学史而不提弗洛伊德的名字，那就不能自诩是一部心理学通史。”精神分析学说在西方心理学界占有举足轻重的地位，被称为心理学第一大势力。而且作为一种重要的社会人文思潮，它对整个20世纪的西方文化也产生了不可估量的影响。单就管理心理学领域而言，“西方所有的管理心理学家实际上都把弗洛伊德学说当作自己观点的理论根源”，尤其是他关于“本能”、“超我”、“人格发展”和“群体心理学”等方面的论述，对管理心理学发展影响深远。

弗洛伊德（S．Freud）是精神分析学派的创始人，精神分析心理学是在弗洛伊德学说基础之上形成的一个心理学思想体系，其主要特点是探寻深藏于人类内心深处的动力源，以及揭示这些动力的性质、强度和活动模式。

精神分析学派可分为早期的经典精神分析学派和后期的社会文化学派。早期的经典精神分析学派主要是指弗洛伊德的精神分析学说，以他的人格理论为核心。其人格理论主要包括人格结构、人格动力以及人格发展三个部分。关于人格结构，弗洛伊德早期把人格划分为“潜意识”、“前意识”和“意识”，而潜意识被认为是三者当中最重要的，是人的精神的动力来源，处于心理深层，主要包括两个方面：一是各种本能冲动，二是被压抑的心理

活动。后来弗洛伊德在《自我与本我》一书中对之前的理论做了修正，提出了本我、自我和超我的三层次人格结构：本我是生物的“我”，依据本能而行动；自我是现实的“我”，依据现实的要求而行动；超我是理想的“我”，按照道德原则和理想而行动。在人格动力方面，弗洛伊德认为驱动人行动的心理能量都来自于本能，一个人做什么或不做什么主要取决于本能在人格结构中的配置和流动。本能在人格三大系统中的分布和转移形成了人格的动力系统。关于人格发展，弗洛伊德认为所有人的人格发展都源于童年的经验，特别是心理性欲发展的经验。

后期的社会文化学派以荣格（C. G. Jung）、阿德勒（Adler）、霍妮（Karan. Homey）和弗洛姆（E. Fromm）等人的观点为代表。他们的共同点在于均不同意弗洛伊德过于强调性本能和其他生物性冲动的观点，而是强调文化的、社会的和心理的因素对人的人格和心理发展的影响。例如，荣格强调历史文化因素的作用，提出了“集体潜意识”学说；阿德勒和霍尼注重心理层面的分析；弗洛姆关注社会文化因素的影响。

精神分析心理学对管理心理学的影响主要表现在以下三个方面：

（1）提出了组织情景中观察人、理解人的一种新视角。作为一种探寻人类潜意识心理世界的理论体系，它超越了传统心理学设定的研究领域和范围，为深入地揭示组织行为的基础和来源提供了新的可能性。

（2）提供了一种新的研究方法：精神分析心理学不仅仅是一套理论体系，还是一套研究方法，它所倡导的自由联想、投射测验技术和日常行为分析方法，均可以应用于管理理论和实践中去。

（3）从心理学的角度探索人类行为的动力系统，弥补了管理学中人性假设研究过于表面化和现象化的缺陷，有助于更加深入地解释人类行为的动力基础和机制，对完善管理心理学中有关的激励理论有重要的启示意义。

二、行为主义心理学

行为主义心理学是20世纪初起源于美国的一个心理学流派，被称为“心理学的第二大势力”，其创始人为美国心理学家华生（J. B. Watson）。1913年，华生在美国《心理学评论》杂志上发表了《行为主义者心目中的心理学》一文，宣告了行为主义的诞生。行为主义心理学认为，心理学不应该研究意识，而应该只研究行为。所谓行为就是有机体用以适应环境变化的各种身体反应的组合。行为主义心理学的影响范围广，持续时间长，在上个世纪60年代以前一度成为心理学的主流研究“范式”。行为主义心理学的发展经历了三个阶段。

（一）早期的行为主义

早期的行为主义以华生为代表，主张心理学应该摒弃一切心理的概念，只研究可以观察和测量的外显行为。心理学只关心引起行为的客观条件和这种条件导致的行为本身，而不关心内部发生的过程，只有这样才能使心理学走上客观实证的道路。华生的行为主义观

点可追溯到桑代克（Thomdike）的联结律和巴甫洛夫的经典条件反射原理。桑代克的联结理论为行为主义搭好了桥，巴甫洛夫的经典条件反射原理为行为主义理论提供了生理学的最终依据。在华生看来，人类行为完全来自学习，尤其是早期训练，而学习的决定因素是外部刺激。外部刺激是可以控制的，控制的基本途径就是巴甫洛夫的条件反射机制。华生指出，复杂的行为是由一系列简单反应联结而成的，而联结的关键是先行反应所产生的动觉刺激成了后继反应的条件刺激。不论多么复杂的行为，都可通过条件反射机制建立起来。

（二）新行为主义学派

新行为主义学派以赫尔（C.L.Hull）、斯金纳（B.F.Skinner）、托尔曼（E.C.Tolman）为代表。

赫尔反对华生将有机体的所有行为都归因于后天学习的极端思想。他以适应性行为作为研究的重点，指出有机体在进化过程中产生了两种不同、但又密切相关的适应手段：一种是在神经组织中固定下来的不学而能的感受器—效应器联结，它是一种对经常发生的紧迫情境做出适当反应的行为机制；另一种是进化过程中习得的行为机制，即有机体本身主动获得的感受器—效应器联结，这便是学习。此外，赫尔认为除了有机体主动学习之外，强化也是行为形成的一种机制。

斯金纳在应答性行为和操作性行为之间做了区分。他认为条件反射有两种：一种是巴甫洛夫的经典性条件反射，主要关心 S-R（刺激—反应）联结是如何形成的，这种条件反射形成过程中有机体只能做出被动的、应答性的反应；另一种是斯金纳自己提出来的操作性条件反射，主要关心 R-S（反应—刺激）联结是如何形成的，这种条件反射形成过程中有机体先做出反应，而后出现刺激，其反应是主动的、操作性的，带有探索或工具的性质，所以又称为工具性条件反射。所谓操作性行为就是这种作用于环境从而产生结果的行为，在这里行为是获得刺激的手段。在斯金纳看来，操作性条件反射原理对于说明人类的现实行为更为适当。如果一个操作性行为发生了，接着就出现了一个正面的强化刺激（如奖励、表扬等），那么这个操作性行为以后出现的频率就会增加；相反，如果没有出现任何刺激或出现的是一个负面的强化刺激（如惩罚、批评等），那么这种行为以后出现的频率就会降低。基于这种原理，只要我们为人类可观察的行为反应设置好恰当的刺激，就可以有效地控制行为，而无需考虑心理内部发生了什么。在这里，斯金纳坚持了华生的环境决定论的基本立场。

托尔曼认为，有机体不是简单地对刺激做出反应，它的行为总是有目的的。在有机体的行为与其要达到的目的之间，必须有一个中介变量，这就是个体的认知。他提出了认知地图的概念，指有机体头脑中预存的周围环境的认知结构。在特定的刺激条件下，有机体就是借助于头脑中认知地图的导引而将行为指向特定目的的。也就是说，在托尔曼那里，传统的 S-R 公式加进了一个中介变量（O），使行为产生的模式成为 S-O-R。

（三）新的新行为主义

新的新行为主义以班杜拉（Albert Bandura）的社会学习理论为代表。其特点在于，一方面将行为主义理论拓展到了社会行为领域，主要关心社会行为的习得过程，另一方面提

出了一整套不同于传统行为主义的新的概念。班杜拉用实验法研究儿童的观察性学习，即在控制条件下，让儿童观察别人（榜样）的行为，然后看他在类似情境中是否模仿榜样的行为。他认为人类许多复杂的社会行为都是通过对榜样行为的观察模仿而习得的。通过这种方式习得的行为不是通过外来强化而是通过内部强化而得以巩固和加强的。他所建立的行为形成模式可以概括为以下的流程图：

榜样 → 观察 → 模仿 → 强化

行为主义心理学对管理心理学的影响表现在如下几个方面：

（1）强调研究内容和方法的客观性，主张用客观的行为观察代替主观的意识内省，这对于组织心理和行为研究科学化和可操作化具有一定的借鉴意义。

（2）行为主义心理学研究的目标在于预测和控制人的行为，斯金纳提出的操作性条件反射原理及强化程式已经被广泛地应用于教学和行为矫正等领域，对于管理活动中薪酬和奖励系统的设计也具有明显的指导意义。

（3）班杜拉的社会学习理论说明了观察学习的重要性及其机制，这对于组织管理中如何利用榜样的力量来引导和鼓励正确行为具有启示意义。榜样的作用还表现在员工培训过程中，如可以为受训者提供明确的榜样供其效仿，消除不当观念和行为，促进学习效果。

三、人本主义心理学

人本主义心理学在 20 世纪 50～60 年代兴起于美国，以马斯洛（A.H.Maslow）、罗杰斯（Carl Rogers）等人为代表，被称为“心理学的第三大势力”。

人本主义心理学反对精神分析学派将人生物学化和行为主义心理学将人机械化和非人格化的倾向，强调人的潜能和价值，认为人生来具有向上向善的一面，有追求自我价值实现的共同趋向。按马斯洛的观点，个体成长发展的内在力量是动机，而动机是由多种不同性质的需要组成的。他认为人有五种基本的需要，分别是生理需要（physiological need）、安全需要（safety need）、爱与归属的需要（love and belongingness need）、尊重需要（esteem need）和自我实现需要（elf-actualization need）。五种需要是逐渐递进的关系，只有满足了低层次的需要，高层次的需要才会显现。每一层次的需要及其满足的程度将决定个体人格发展的水平。在心理学上，需要层次论是解释人格发展的重要理论，也是解释动机变化的重要理论。罗杰斯则将人本主义思想运用于教学和临床实践，提出了“以患者为中心的疗法”和“以学生为中心的教学法”。

人本主义心理学对管理心理学的影响是巨大的，主要体现在以下几个方面：

（1）把人的潜能与价值提高到了心理学研究的首位，突出了人的动机系统与高级需要的作用，这本身对于管理实践就具有现实指导意义。马斯洛很早就关注人的因素，当美国 20 世纪 60 年代盛行科学化管理、工作设计等做法时，他就提出工业领域也许是心理动力

学和高级人类动机研究的生态学实验室。

（2）促进了以人为本的组织管理和心理治疗的发展。在倡导人性化管理的今天，人本主义心理学的现实意义尤为明显。马斯洛很早就认识到了人本管理的重要性，他指出任何组织的管理问题都可以用一种新方法来加以解决，建立起某种环境条件，使个人目标与该组织的目标结合起来，即无论何种管理都应以人为本。

首先，他尊重个人，强调自我实现，主张以最简单的方式，对人类劳动、生活和谋生的方式进行适当的管理。适当的管理是一种理想化的或革命性的技巧。

其次，他倡导健康管理和协同管理，主张改变在大型组织里一直在起作用的那种顺从性行为模式，让人们在一个公司里保持自己的个性，使工作不成为一种工作，而成为自己喜欢的娱乐。

第三，他提出了协同发展的观点，认为一个追求自身利益的人同时也自动地帮助了别人，而无私奉献帮助他人的人同时也满足了自我需要。

四、认知心理学

认知心理学有广义和狭义之分。广义的认知心理学不仅仅指信息加工心理学，还包括以认知过程为主要研究对象的各种心理学流派和理论，如格式塔学派、皮亚杰的认知发展理论等。狭义的认知心理学专指信息加工心理学，又称现代认知心理学。

20 世纪 50～60 年代，由于心理学中占统治地位的行为主义发生危机，以及由于信息论、控制论、计算机科学和语言学等新兴学科的影响，发生了一场心理学的运动，其结果是产生了现代认知心理学，亦称信息加工心理学。它反对行为主义心理学无视人的内在认知活动的环境决定论和机械决定论，注重研究人类的内部认知活动，强调人们认知活动的复杂性、多样性、主动性和能动性。它深受当代计算机科学和信息科学的影响，把人脑类比为计算机，用计算机科学的概念和原理来说明和解释人的认知过程。在研究方法上，它既倡导实验的或实证的路线，又接受传统的内省方法，还创立了计算机模拟等新技术，呈现了兼容并包的趋势。

现代认知心理学一经产生，便迅速占领了心理学研究的各个领域，成为当代心理学研究的一种新的范式。迄今，它已经在诸多领域取得了丰硕的研究成果，产生了广泛的影响。

在管理心理学领域，认知心理学的影响主要表现在如下几个方面：

（1）主张研究人的内部认知过程，强调人的认知活动的主观能动性和复杂多样性，这就从一个侧面突出了管理过程中人的因素的重要性，特别是突现了管理者和员工的聪明才智和创造性的作用，对于改变传统的以“管”为主的管理模式具有启示意义。

（2）认知心理学的诸多发现对于管理实践具有直接或间接的指导和启示作用。了解和掌握人们认知活动的特点和规律，按照这些特点和规律开展管理活动，对于管理的科学化、柔性化具有举足轻重的意义。

（3）认知心理学兼容并包的方法论和方法，对于开展管理心理学研究和解决实际的管理问题也具有重要的指导意义和参考价值。

本章小结：

1．管理学科的基础理论与心理学科的基础理论一起构成了管理心理学成长发展的理论基石。人性理论、博弈理论从管理学的角度为管理心理学奠定了理论基础，而精神分析心理学、行为主义心理学、人本主义心理学和认知心理学则从心理学的角度为管理心理学的发展铺平了道路。

2．从传统的经济人假设发展为社会人、自我实现人、复杂人假设，反映了西方管理学对人性认识的发展过程，标志着管理思想的进步。继四种人假设之后提出的文化人、决策人假设，是对人性认识的进一步深化。

3．博弈论被称为 20 世纪人类最杰出的贡献之一。博弈论在激励和约束员工、绩效考核等方面充实了管理心理学的内容。

4．管理心理学的发展，始终受到不同研究取向心理学理论的影响，其中尤以精神分析心理学、行为主义心理学、人本主义心理学和认知心理学为重。它们为管理心理学学科大厦的建构做出了各自独特的贡献。

本章思考题：

1．阐述各种人性假设的基本观点及其相应的管理措施，比较这些假设。

2．你认为哪种人性假设比较切合实际？为什么？并举例说明。

3．博弈论的主要观点是什么？阐述博弈论对管理心理学的影响。

4．管理心理学有哪几种心理学的基础理论？分别阐述这些理论对管理心理学的影响。



第二篇　动机管理

第四章　动机与管理

动机是人类一切行为的出发点和原动力，组织行为也不例外。在组织情景中，员工做什么或不做什么、是主动积极地做还是被动消极地做、愿意投入多大精力和坚持多长时间来完成组织任务和实现组织目标，无不取决于他的动机。因此，研究和探讨员工动机变化的特点和规律、开展动机管理是其他一切管理活动的前提，也一直是管理心理学关注的中心问题之一。在管理心理学中，动机问题始终与员工激励问题联系在一起，既是员工激励的出发点，又是员工激励的目标和归宿。本篇将用三章内容较为系统地阐述动机及与之相关的激励问题。

在本章中，首先讨论动机的一般概念、功能及其与行为效率之间的关系，然后介绍动机理论的历史演变及常见的动机类型，最后探讨动机的激发和调适问题，以便为后面关于激励理论及其应用问题的讨论做好理论上的铺垫。

第一节　动机的一般概念

一、动机的涵义

动机是一种非常复杂的心理现象。在管理心理学中，动机（motivation）是一种由目标引导、激发和维持个体活动的内在心理过程或内部动力系统。也就是说，动机是由一定的目标引导和激发的，它产生原动力推动个体的行为。而这种原动力经由个体对目标的认识，将外部的诱因变成内部的需要，进而推动个体去从事某种活动。

图 4.1 形象地说明了动机产生的的心理过程——需要唤起了指向诱因的驱力。从系统论的观点来看，动机包含以下三个既相互作用又相互独立的因素：



图 4.1　动机产生的心理过程

（1）需要：当个体在生理上或心理上产生不平衡的时候，需要就产生了。例如，当我们感到饥饿、口渴时，会产生饮食、饮水的需要；而当我们因缺少朋友和伙伴而倍感孤独时，就会产生交际的需要。

（2）驱力：在动机产生的心理过程中，驱力处于非常核心的位置，它为指向目标的行为提供了原动力，并将个体的生理和心理需要维持在一定的水平上。例如，对食物和水的需要转化成了寻求食物与水源的驱力，对朋友的需要转化成了交往的驱力。

（3）诱因：诱因是动机过程的最后一部分，它是指可以减轻需要程度、降低驱力水平的外部对象。诱因可以重新恢复个体生理或心理上的平衡，降低或消除驱力的影响。例如，食物、水和朋友都可以使我们恢复平衡，减轻相应的驱力。在此，食物、水和朋友都是诱因。

二、动机的作用与功能

作为行为活动的一种内部动力，动机具有以下四种功能：

（1）激发功能。动机是行为的“发动机”。人的各种各样的活动总是由特定的动机引发的，不存在没有动机的行为。正如汽车的发动机可以让汽车由静止状态变化到运动状态一样，动机可以让个体从静止状态变化到活动状态。例如，在企业中，某员工希望自己在年终取得更好的业绩，在这种动机的驱动下，他就会产生相应的行为（如努力工作、维系好人际关系等）。

（2）导向功能。动机是行为的“方向盘”。它不仅能激发和引起行为，还能将个体的行为引向特定的目标或对象，使个体的行为朝着特定的方向前进。例如，在成就动机支配下，人们就会主动选择具有挑战性的工作；在权力动机驱使下，人们可能试图去影响和控制他人和周围环境等。动机的导向功能赋予人的行为特定的内涵和内容，使其有了正确与错误、高尚和卑劣之分。

（3）维持功能。动机引起某种行为并将其导向特定目标之后，还需要继续发挥其作用，即维持已引起的活动，直到实现目标为止。例如，具有较强成就动机的管理者，在追求成就目标的过程中，无论遇到什么困难与挫折，都应该坚持自己的想法和行动，直到达到预期目标。若非如此，哪怕当初的动机多么强烈，目标多么明确具体，也只能半途而非，无功而返。

（4）调整功能。动机是行为的“变速器”。在为个体行为提供动力与方向的同时，又对人的活动进行控制与调整。例如，一个具有较强学习动机的学生，当他现阶段进行的活动与该动机的目标活动之间出现偏差时，他的学习动机就会自发地调整其现有的行为，使之朝着预定目标不断前进。

三、动机与行为效率

（一）动机与行为的一般关系

一般说来，人们的行为总是由一定的动机引起的，动机也总是要表现为特定的行为，

但动机和行为之间却存在着复杂的关系，具体表现在以下几点：

（1）有动机不一定有行为，因为行为的发生还需要其他因素，如内心对某物某事的需求程度、行为合理性的权衡、外界的客观条件等。

（2）相同动机可能表现为不同行为，如在成就动机驱使下，个体可能表现出一系列的行为，诸如努力工作、敢于冒风险、接受挑战性的任务等。

（3）相同行为可能由不同动机引起，如员工努力工作可能受高收入、高社会地位、周围人的认可等外部动机的影响，也可能受来自内心的责任感、工作上的完美倾向等内部动机的影响。

（二）动机与工作效率

由于动机与效果并非总是统一的，现实生活中常常出现有动机无效果或有效果无动机的情况，所以在判定人们的行为时，动机和效果孰轻孰重的问题一直是困扰许多领域专家的理论和现实问题。例如，在对人们的行为进行道德判断或对犯罪嫌疑人的行为量刑定罪时，这个问题始终未得到很好的解决。在组织管理活动中，也存在类似的问题。例如，在评价员工的行为和考核员工的绩效时，究竟是重态度、重动机还是重行为、重效果，也始终困扰着管理者们。鉴于这个问题是一个带有理论探讨性质的大问题，这里不予展开讨论。

与管理心理学密切相关的一个问题是动机水平或强度对工作效率的影响问题。人们通常认为，动机越强，个体活动的热情越高涨，活动的次数越频繁，活动的效果也越好。但事实并非如此。个体工作动机较低，会使其对工作不热情、不积极，从而导致工作效率的低下，这固然是不好的。但如果动机过于强烈，个体就会处于高度紧张的状态，其认知思维活动就会变得僵硬、狭隘、不灵活，也会影响能力的正常发挥，从而降低工作效率。例如，一位员工即将参加一场培训结果的测试，如果他希望通过的动机过于强烈，就会出现紧张和焦虑的反应，这会严重阻碍其记忆、思维或动作技能的正常发挥，最终影响测试结果。那么，究竟动机水平多高时最有利于提高工作效率呢？

心理学家耶克斯和多德森（Yerkes & Dodson，1908）的研究对此问题给出了答案。他们的研究结果如图 4.2 所示。该图表明：各种活动都存在一个最佳的动机水平。动机不足或过分强烈，都会使工作效率下降。而且，动机的最佳水平随着任务的难度和性质的不同而不同。在比较容易的任务上，工作效率随动机水平的提高而上升，总体上需要的最佳动机水平较高；在难易适中的任务上，中等水平的动机可以使工作效率达到最佳；随着任务难度的增加，所需要的动机水平有逐渐下降的趋势，也就是说，在困难和复杂的任务上，较低的动机水平最有利于任务的完成。这就是著名的耶克斯—多德森定律（Yerkes-Dodson Law）。按照这个定律，管理者需要根据任务的难度水平和复杂程度适当地调整员工的动机水平，特别是在那些难度高、所需要的动作或心智技能复杂的任务上，反而应该适当地降低员工的动机水平。

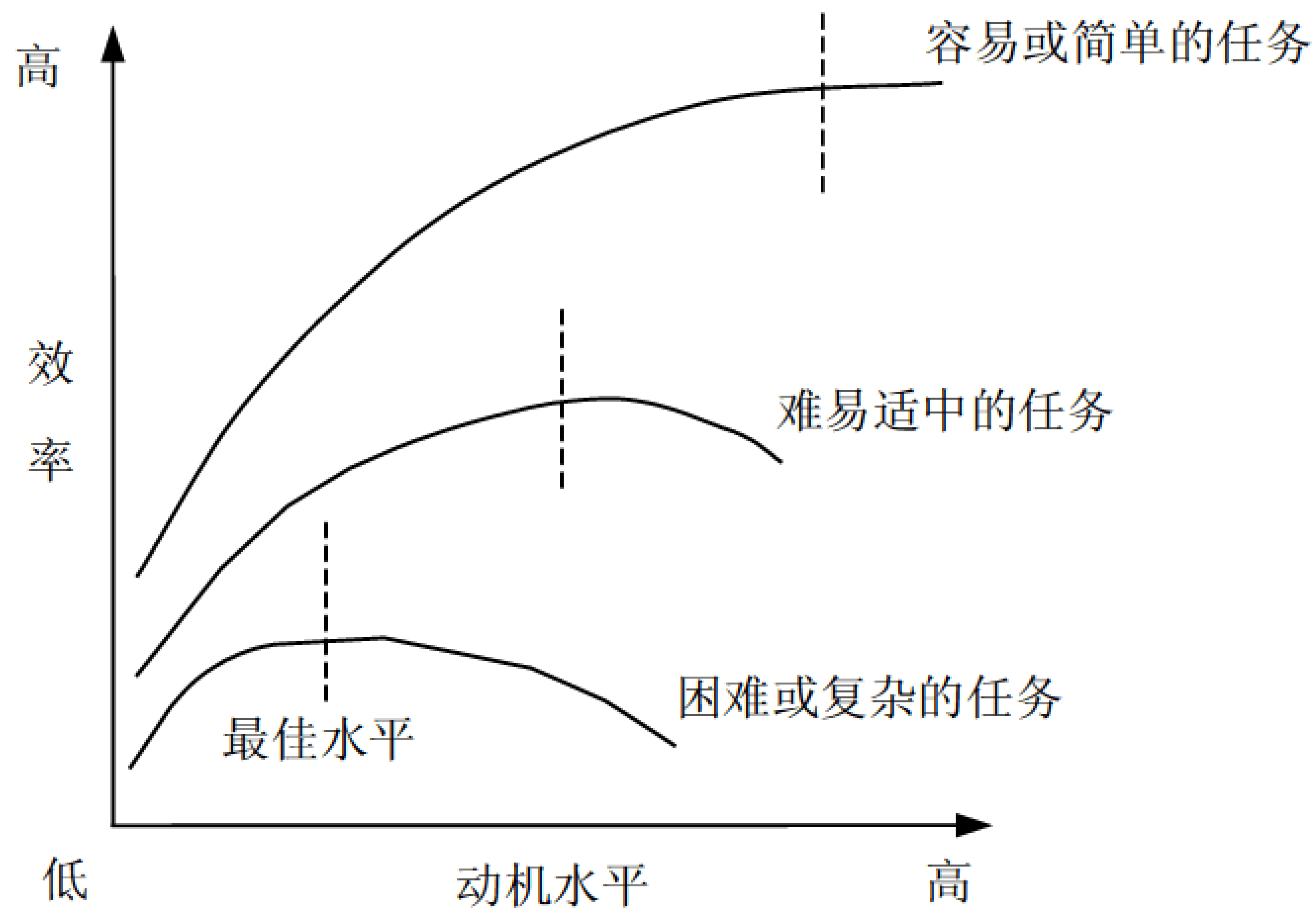


图 4.2 耶克斯—多德森定律

第二节 动机研究的历史演变

一般说来，心理学中的动机研究以 20 世纪 60 年代为界可分为两个时期，之前主要以行为主义和精神分析理论为主导，强调本能、冲动、驱力等生物性因素在决定人的动机和行为方面的直接作用，因此称这一时期出现的各种理论为生物因素的动机理论；20 世纪 60 年代以后，认知的观点逐步介入到动机研究中来，相应地称之为认知因素的动机理论。下面就对这两大类理论加以介绍。

一、生物因素的动机理论

一般说来，生物因素的动机理论主要包括早期的本能理论、驱力理论和诱因理论等。这些理论的共同特点为：都研究动机行为的生理结构和机能，试图从有机体的本能、冲动、驱力等生物性因素来寻求人类或动物行为背后的动因。

本能理论认为人类行为的根源和动物一样，都受本能的驱使。所谓本能，是先天的遗传倾向，是物种在进化和适应环境的历程中形成的，它是一种不学而能的行为模式。本能是人类一切思想和行为的基本源泉和动力，个体的种种行为表现都可以归结为本能反射。本能理论的主要代表人物有麦独孤、弗洛伊德等心理学家。这种理论过分强调人的生物性，而忽视了社会性，把人等同于动物，因而是较为片面的。

随后的驱力理论则用驱力的概念来代替本能的概念。所谓驱力是指个体由生理需要或心理需要所引起的一种紧张状态。这种理论认为有机体的生理或心理驱力是人类行为的动

因，它能激发或驱动个体行动以满足需要，消除紧张，从而恢复机体的平衡状态。赫尔是驱力理论的主要代表人物。驱力理论仍以人的生物性本能为出发点来解释人类的行为，因此也摆脱不了机械性的局限。

与上述两种理论不同，诱因理论主要从有机体的外部来寻求行为的动因，强调外界的诱因在行为产生中的作用。所谓诱因是指能满足个体需要的刺激物，它具有激发或诱使个体朝向目标的作用。诱因是个体行为的一种能源，是使行为持久的推动力。这种理论的代表人物主要有巴甫洛夫和斯金纳。

纵观上述生物因素的动机理论可以发现，这些理论的贡献在于，它们能够从动机的内在起因和外在诱因两方面对有机体的行为进行一定程度的解释，其不足在于把诸如本能、冲动、驱力等内在的生物性因素作为人类行为的动因，而忽略了其他因素的重要作用。事实上，人类行为背后的原因是相当复杂的，仅用生物因素是难以加以解释的，于是就出现了一系列认知因素的动机理论，使动机研究进入一个新的时期。

二、认知因素的动机理论

认知因素的动机理论的共同点是：都认为人是理性的，强调人的认知因素或变量在行为动机产生中所扮演的中介调节作用，这些认知因素包括人们对环境的分析、对目标的预期或期望、对成败可能性的分析等。该理论已经在教育、心理治疗、管理等领域得到了广泛的应用，具有较高的实用价值。在此，仅介绍其中与组织管理情境联系较为密切的两种理论。

（一）动机的认知一致性理论

认知一致性理论是一种比较典型的认知因素的动机理论，该理论的基本假设是：个体认知（思想、态度等）的不一致能产生动机，从而激发个体努力减少或消除这种不一致，以此来寻求认知的协调或一致。也就是说，人的社会认知过程是一个由平衡到不平衡再到平衡的动态过程，通常个体的心理处于平衡状态，一旦出现认知矛盾则会打破这种心理平衡，使主体感到压力与不快，在保持心理平衡的需要与消除不快情感需要的驱使下，个体就会采取行动使其认知系统重新恢复平衡。认知一致性理论包括认知平衡理论和认知失调理论。

1．认知平衡理论

心理学家海德（F.Heider）在 1946 年提出了认知平衡理论。他认为人们喜欢或倾向于使自己对人或事物的认知达到平衡，认知不平衡则会产生争取平衡的行为动机，从而寻求各种方式把不平衡转化为平衡。据此，海德提出了 P-O-X 模型，其中 P 代表个人，O 代表另外一个人，X 代表一些目标对象（如情境、事件、思想等）。海德认为，项目（P、O、X）如果有归属的关系，就构成一个单元。如果该单元在各方面具有相同的动力特性，那么它就处于一种平衡的状态，不存在去改变这种状态的压力。但如果单元中各个项目不能和谐

并存，就会引起个体的紧张与不安，从而产生获取平衡的动机。举例来说，某男生 P 喜欢某女生 O，而 O 又爱好音乐，假如 P 对音乐无兴趣，则 P 会产生认知上的不平衡，这种不平衡会驱使 P 做出某种努力来扭转这种认知的不平衡。由此可见，认知的不平衡确实能成为行为的动力，起到动机的作用。

2．认知失调理论

认知失调理论是由费斯廷格（L.Fstinger）于 20 世纪 50 年代后期首先提出来的。所谓认知失调是指个体所持有的认知彼此矛盾，处于相互对立的状态。例如，“我很喜欢体育”和“母亲不准我参加体育活动”这两种认知，便是失调的。在费斯廷格看来，认知失调将会造成个体心理上的不愉快和紧张，从而驱使其采取某种行动以减轻或消除认知失调。一般说来，失调强度越大，人们想要减弱或消除失调的动机就越强烈，而失调程度的大小主要取决于两个条件：首先，失调的程度同某一认知要素对个体的重要性成正比，例如，丢掉一元钱的失调程度要远远低于丢掉一份满意的工作所引起的失调；其次，失调程度还依赖于失调的认知数目与协调的认知数目的相对比例。这两个条件与认知失调程度之间的关系可以用下面的公式来表示：

$$\text{失调程度}=\frac{\text{失调的认知数目}\times\text{认知要素的重要性}}{\text{协调的认知数目}\times\text{认知要素的重要性}}$$

此外，费斯廷格在大量研究基础上提出了减小认知失调的三种方式：（1）改变某一认知因素，使其与其他因素之间失调的关系趋于缓和。例如，“我喜欢抽烟”和“抽烟有害健康”是不协调的两个认知因素，个体就可以通过减少抽烟量或搜集抽烟无害于健康的证据来缓和认知协调。（2）突出某一认知因素的重要性，使其“战胜”另一认知因素。例如，可以特别强调抽烟的危害性，在心理上产生恐惧感，彻底戒烟，从而达到认知的协调。（3）增加新的认知因素来协调已有认知因素之间的关系。例如，通过增加 “肺癌只要早发现，及时治疗，无生命危险”这一新的认知因素来降低失调。

（二）动机的归因理论

动机的归因理论主要是由心理学家韦纳（Weiner.B）提出来的，他从归因的视角来研究人类行为背后的动因。依照这种理论，当个体面对成就情境中某些特定的行为结果时，他会有意无意地寻找其产生的原因，找到的原因不同，个体随后的动机水平也就不同。韦纳通过对各种原因进行逻辑分析，最后总结出原因的三种特性或对原因进行分类的三个维度：部位、稳定性和可控制性。其中每一个维度都衍生出了两类不同的原因，依次分别是：内部原因－外部原因、稳定原因－不稳定原因、可控制原因－不可控制原因。例如，个体的努力程度是一种内部的、稳定的、可控制的原因，而运气就是一种外部的、不稳定的、不可控制的原因。原因的各种特性及其之间的关系如图 4.3 所示。

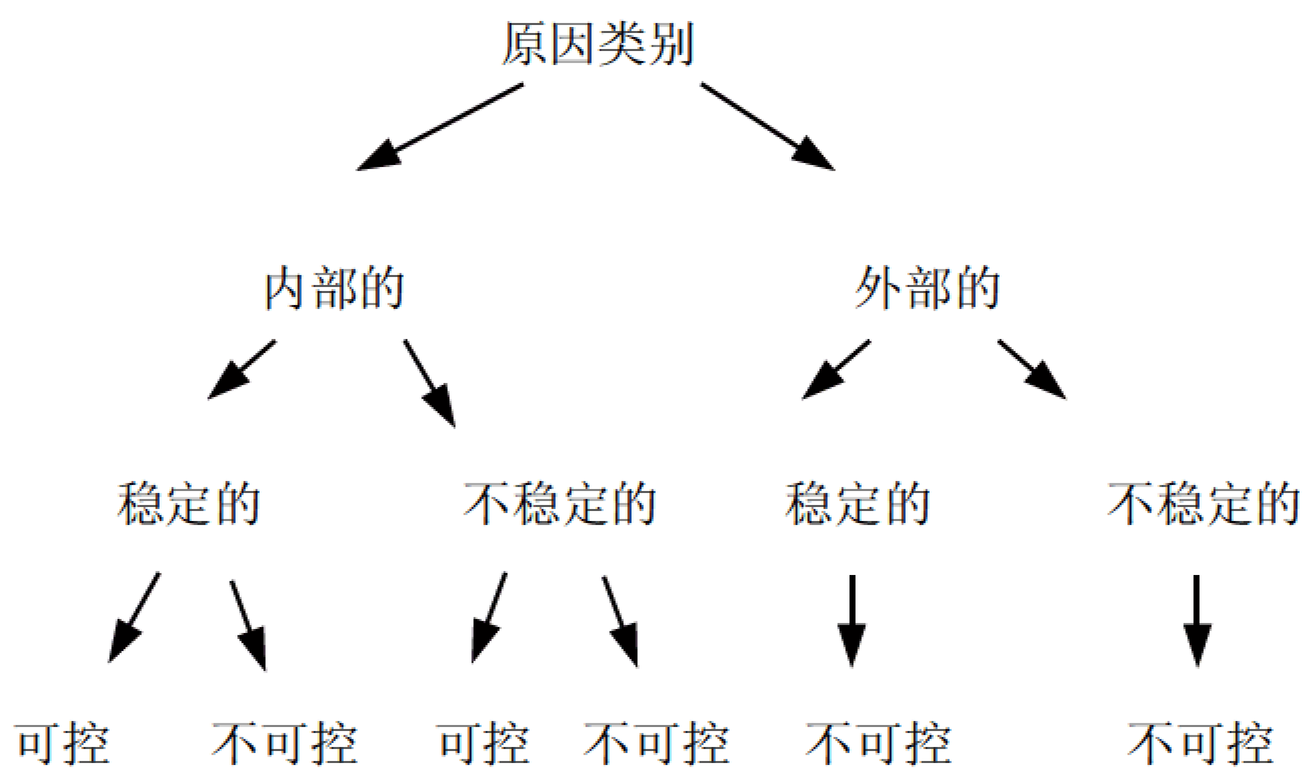


图 4.3　原因特性的三个维度

那么，人们对行为的归因怎样影响其随后的动机呢？韦纳认为，归因是通过影响人们对未来的预期或期望及情绪、情感，进而影响人们动机的。在此，把动机随归因不同而发生变化的规律总结如下：（1）当失败被归因于稳定的、内部的、不可控制的原因时，将会弱化进一步活动的动机，而失败被归因于不稳定的、外部的、可控制的原因时，则不会弱化甚至还会强化进一步活动的动机；（2）当成功被归因于稳定的、内部的、可控制的原因时，将会强化进一步活动的动机，而成功被归因于不稳定的、外部的、不可控制的原因时，则无助于强化甚至会弱化进一步活动的动机。例如，在企业组织中，如果一名员工将自己较差的工作业绩归因于能力低这种稳定的、内部的、不可控制的因素，他可能会感到无助和无望，采取破罐子破摔的消极态度，失去进一步努力的动力；反之，如果他将不良的业绩归因于努力不够这种不稳定的、可控制因素，他就不至于对未来产生无助和绝望的感觉，甚至还会主动和上级沟通，重整旗鼓，努力提高自己的工作绩效。由此可见，对行为结果的归因确实会影响到个体随后的动机。作为管理者，在员工面对成功或失败的结果时，如何引导其作出合理的归因，既是动机管理水平的体现，也是管理能力的重要体现。

在韦纳动机归因理论的基础上，后来又派生出了两种颇有影响的动机理论：自我效能理论和习得无助理论。

自我效能理论是心理学家班杜拉（Albert Bandura）提出来的。自我效能通常是指个体对有效地控制自己生活诸方面的能力的知觉或信心。动机的自我效能理论的中心思想是：个体的自我效能感决定他在成就情境中的行为动机。具体来说，自我效能感高的人在有关的活动中行为积极性高，乐于付出努力和采取策略来应付遇到的问题、解决面临的困难，而当问题和困难得到解决和克服的时候，他当初的效能感就得到了证实，这就维持了动机。即便当个体偶尔遇到前所未有的困难时，他对有能力取得成功的信心也有助于克服先前操作的消极方面，诱发动机行为。相反，自我效能感低的人在有关的活动上行为积极性低，不愿付出过多的努力和采取相应的策略应付困难、解决问题，这必然导致活动结果不尽如人意，反过来又降低了他的效能感。而个体的自我效能感不仅仅来源于他所遭遇的经历和

处境，更重要的是取决于他对自己遭遇和处境的认识，特别是归因。按照这种理论，在组织管理中，如何通过各种途径和方法培养和提高员工的工作自我效能感，是激发员工工作动机、点燃其工作热情的关键，其中一种重要途径就是引导员工对其成败工作绩效作出正确归因。

习得无助理论首先是由塞里格曼（Martin E.P. Seligman）根据动物实验的结果而提出来的。将老鼠放在一个有电击的区域，起初它尝试各种办法、作出种种努力试图逃离电击区域，后来发现无论如何做也无法逃离，最后就再也不做任何努力了，也就是说它习得了一种无助感。后来研究者们将习得无助的概念推广到人身上，是指当人们作出种种努力都无法改变某种行为的结果时习得的一种行为与结果不相倚的信念，这种信念让人失去了对结果的控制感，一旦固定下来，后来遇到类似问题时，也会由于内心的无助感而放弃任何努力。很显然，习得无助理论是关于动机减弱或缺失的一种理论。那么，难道作为具有理性分析能力的人真的会像动物那样仅仅因为不断遭遇挫折而习得无助吗？人们对行为结果原因的知觉和分析在无助感的产生中究竟起什么作用呢？基于这样的考虑和问题，在习得无助理论后来的发展中逐渐吸收了归因研究的一些思想和成果，引入归因变量来解释人类习得无助的形成和变化规律。基本的观点是：如果一个人将不可控制的消极结果归因于内部的、稳定的、普遍的因素，那么一种弥漫性的无助或抑郁状态就会出现，自我评价降低，动机减弱到最低水平；如果不是这样，那么人们在经历不可控制的消极事件后所产生的无助感就只限于特定的时空条件，不足以降低人们的自我评价和弱化人们的动机，甚至还会强化随后行为的动机。这就是说，消极事件原因的可控制性和部位、稳定性、普遍性一起决定了无助感的产生和随后的行为动机，而不是单独起作用的。在组织情境中，管理者需要通过建立完善的反馈和激励系统，并引导员工对失败和挫折进行积极的自我归因，来防止员工习得无助感的产生，避免员工出现自暴自弃、破罐子破摔的消极心态。

总之，无论是生物因素的动机理论还是认知因素的动机理论，都从各自独特的视角探究了个体行为背后的动因。每种理论都不可能尽善尽美，都有其合理性和局限性。鉴于此，当应用这些理论来解释组织情境中人们的行为时，就应该取每种理论的精华而去其糟粕。管理者应在充分了解有关动机理论的基础上，有效推测员工行为背后的动因，提高对员工行为的预见性，为激励、考核评估等管理工作奠定良好的基础。

第三节　动机的种类

动机是一种复杂的心理现象，从不同角度和层面可以把动机分为不同的种类。在本节中，首先介绍动机的一般分类，随后重点阐述与组织情境密切相关的几种社会性动机，如成就动机、权力动机、亲和动机和工作动机等。

一、动机的一般分类

（一）生理性动机和社会性动机

根据动机的性质，人的动机可以分为生理性动机与社会性动机。

生理性动机是指个体为满足生理、本能需要而产生的动机。例如，饥饿、干渴、睡眠等动机，都是生理性动机。这种动机具有原始性、自发性，是有机体与生俱来的，并且为人和动物所共有。但是由于人是社会的实体，所以纯粹的生理性动机是很少的。社会性动机是指由个体的社会物质、精神生活需要而引起的动机。例如，权力动机、成就动机、审美动机等。这种动机为人类所特有，是个体在先天倾向的基础上，在广泛的社会实践活动中形成的，具有社会历史性。后面将会详细介绍几种常见的社会性动机。

（二）外部动机和内部动机

根据动机的来源，可分为外部动机和内部动机。

外部动机是指由外在力量激发而来的动机。在组织环境中存在许多外在诱因，如高报酬、高职位、和谐的同事关系、上司的赞赏等，同时也包括避免惩罚的驱力，例如避免解雇或者调职等。这些都可以吸引和激励员工，成为推动员工积极工作的诱因。内部动机是指由内在心理因素转化而来的动机。内部心理因素包括很多内容，例如兴趣、爱好、责任感、成就感、自我实现需要等，这些心里因素在一定条件下可以成为推动员工积极活动的内部力量。

需要注意的是，这两种动机并不是完全独立的。许多激励措施同时具有激发外部动机和内部动机的双重作用。例如，一位销售人员在销售竞赛中取得了胜利并得到了物质奖励，在这里，竞赛这种激励方式既激发了外部动机（物质奖励），又激发了内部动机（获得“优胜”的成就感）。此种情境中，内外动机就是相互交叉、互相推动的。

（三）主导动机和辅助动机

根据动机在活动中起作用的大小，可以把动机划分为主导动机和辅助动机。

主导动机是指在个体的动机体系中，对行为起调节和支配作用的动机。它是一个人最强烈、最稳定的动机，通常对行为有决定意义，具有更大的激励作用。相反，对个体行为没有决定意义、仅起辅助作用的动机，叫辅助动机。例如，某个组织成员工作积极努力，其工作动机可能有：为实现组织目标贡献力量、引起领导的重视、争取较高的酬金、表现自己的工作能力等。在这些动机中，争取较高的酬金可能是主导动机，而其余为辅助动机。在行为的发生过程中，主导动机起的作用最大，支配着行为发生的方向和强度，但与此同时，辅助动机也起一定的作用，也可能对行为有一些其他方面的影响。

总之，对动机进行分类，目的是从不同的侧面来研究动机的性质、机制以及它在管理活动中的作用。需要说明的是，上述各种分类仅具有相对的意义，而不是绝对的，还可以根据其他的标准来对动机进行分类。

二、社会性动机

（一）成就动机

一般地说，可以认为成就动机（achievement motivation）是驱动一个人在社会活动的特定领域力求获得成功或取得成就的内部力量，在行为上它表现为一个人对自己认为有价值的、重要的社会或生活目标的刻意追求。理解这个概念至少应该注意以下几点：（1）它是后天获得的，而不是与生俱来的。没有哪个人天生就具有很高的或很低的成就动机。（2）它是摆脱了低级需要限制的、具有社会意义的发展性动机，而不是受低级需要驱动的缺失性动机。这就是说，成就动机是一种发自内心的力求发展的高级动机，而不是由生理的匮乏状态引起的低级动机。所谓有成就或无成就、成功或失败都是根据一定的社会标准来衡量和确定的，所以成就动机必然是与特定的社会目标相联系的。一个人所追求的目标离个人的低级需要或趣味越远、社会性越强、社会价值越高，它就越具有成就动机的性质。很难说那些终日忙碌于满足自己低级需要的人有什么成就动机。由于只有人类才会在满足了自己的低级需要的基础上产生较高级的社会性需要，所以可以说成就动机是人类特有的动机。（3）它是个体的，而不是种族的。正由于成就动机是在后天的社会生活中获得的，而不同的个体又有不同的生活环境和道路，所以它必然是因人而异的，具有个体性。一个人成就动机的水平、内容和性质是其个性特征的主要体现，而人与人之间在成就动机的水平、内容和性质上的差异，是人们个性差异的主要表现形式之一。在满足低水平需要的活动上，人与人之间、甚至人与动物之间很难区别开来，即使能够区别开来，也只是一些非本质的、细枝末节的差异，衣食住行、柴米油盐，不过如此。但在满足高水平需要的活动上，人与人之间的一些重要的、本质的差异就能体现出来。

美国哈佛大学心理学家大卫·麦克莱兰（D.C.McClelland）根据人们成就动机的特点，区分出了追求成功的人和避免失败的人。追求成功的人对成功感到骄傲，对失败则不那么感到羞愧；而避免失败的人对失败感到无地自容，对成功则不感到骄傲，这是由他们选择目标的不同造成的。在目标的选择上，追求成功的人通常选择那些中等难度的材料或任务，敢于接受挑战，并对自己的行为结果负责；而避免失败的人所选择的材料或任务不是太难，就是过易，太难的任务可以推卸责任，过易的任务可以避免失败。追求成功的人对未来充满希望，认为在将来的活动上取得成功的可能性比目前正在完成的任务成功的可能性还要大；而避免失败的人则认为目前的情况已经不错了，将来不会比现在更好。

成就动机对个人的发展有重要的意义。许多研究发现，在两个人的智商大体相同的情况下，成就动机高的人比成就动机低的人在活动中成功的可能性要高。在学校里，成就动机高的学生成绩较好，名次较高；在企业里，成就动机高的员工有可能取得较好的业绩。实际上，成就动机不仅影响个人的发展，而且还能推动社会经济的发展。有关社会成员成就动机水平与社会经济状况关系的研究指出：社会群体总体的成就动机水平与其社会经济发展水平之间有着密不可分的关系，如在古希腊由兴盛到衰落的过程中，社会成员动机水

平也表现出了由高到低变化的特点。

另外，成就动机水平的高低还影响人们的职业选择。心理学研究发现，成就动机水平高的个体在为自己选定职业时大多比较现实，能从自身水平出发，选定一个比较实际的职业目标。一项关于密执根大学学生的调查显示，81%的高成就动机的学生对职业的选择比较现实，而在低成就动机学生中，只有 52%的人能够为自己选定一个较为实际的职业。另一项对美国男性所做的调查显示，高成就动机个体对自己的职业有更多的积极评价，主要表现为对工作的满意度较高、对所从事的工作充满兴趣、喜欢工作、在选择自己的职业时较少受家庭的影响等。

最新的研究表明，公司的利润与管理者的成就动机有显著的正相关。研究还发现，高成就动机的管理者更尊重他的下属人员，比低成就动机的管理者更容易从下属那里获得新想法，也更尊重下属的新想法；中高级管理人员的成就动机与他们今后的职位晋升呈正相关。此外，无论是男性还是女性，企业家的成就动机都要显著高于员工的成就动机。成就动机在那些全职的成年员工身上，具有跨国家的普遍性。一项关于不同国家成就动机的比较研究发现，成就导向具有国际差异。具体来说，成就导向在美国这种个人主义的文化背景下是最高的，而在日本和匈牙利这些集体主义取向的国家里是最低的。

（二）权力动机

权力动机（power motivation）是指人们支配和影响他人及周围环境的内在驱力。在权力动机支配下，人们表现出积极主动的参与精神并有成为某一群体领导者的愿望。高权力动机者经常表现为对社会事业的浓厚兴趣，并且会积极地寻求影响他人的方式，例如，他可能会使用上下级关系或外在报酬，来控制其他人的行为，在讨论问题时总是试图以自己的观点、看法去说服别人，在群体中希望处于领导地位，日常生活中表现得比较健谈、好争论等。

管理小贴士：

测量成就动机的方法——主题统觉测验

主题统觉测验（Thematic Apperception Test，TAT）被证明是测量成就动机的一种非常有效的工具，它能够很好地辨别和测量成就动机。测验是这样进行的：其中的一张图画上显示有一位年轻人正在耕田，太阳正要下山。要求被测验的人根据其在图片上看到的内容讲一个故事。故事的内容能够投射出个体的主要动机。例如，如果受测者认为非常遗憾，太阳都快要下山了，但是这个年轻人还有好多的地要耕种，他还想在下雨之前将种子种下去。这样一个反应表明受测者有很高的成就驱力。一个有较低成就动机的人可能会说，这个年轻人很高兴，因为太阳快要下山了，他终于可以回家了，喝上一杯酒好好轻松一下。这种测量成就动机的方法是非常有效的，以至于经常被研究者和专业工作者所采用。

权力动机与组织行为之间有重要的联系。麦克莱兰对权力动机做了相当多的研究，他认为权力动机主要有两种不同的表现形式。第一种表现形式运用了个人支配、生理侵犯和

非法利用。用这种形式来表现权力动机的人，用胜败的观点来看待各种情境。他们必须获胜，而其他人则必须失败。麦克莱兰认为这种权力行为不能够带来所想要的领导方式。

第二种表现形式所关注的是说服力和人际影响力。这种人试图唤起他想要施加影响的人的自信心。例如，向大家澄清群体的目标，说服成员们来实现这个目标，努力让这些成员培养起对自己能力的信念等。麦克莱兰认为这种权力动机表现形式是组织中有效管理者所具备的特征，这种管理者能够带来组织急剧的变革和创新，实现组织的发展目标。

权力动机对取得管理成功的作用得到了美国电话电报公司所进行研究的支持。该公司对不同动机状况的管理人员的晋升率进行了跟踪研究，发现权力动机强烈的管理人员工作绩效一贯较好，进步也快，而且职位也较高。

（三）亲和动机

亲和动机（affiliation motivation）是指个体对于建立并保持良好人际关系、受人喜爱以及与周围人融洽相处的关注。具有很高亲和动机的个体会表现出强烈的与人友好相处的愿望，希望自己能够被周围人所接受和认可，愿意参与到人际交往活动之中。他们在团队或团队之间起到重要的调和作用，因为他们在所有的交往圈中都试图获得和谐亲密的人际关系。相反，具有中等或较低亲和动机的人更有可能对长时间的独自工作与学习感到自在，但他们通常也能够与他人建立并维持良好的人际关系，被团队成员所接受。

亲和动机是个体保持社会交往和人际关系和谐的重要条件，与组织行为有着密切的关系。但值得注意的是，亲和动机也有一个适度的问题。麦克莱兰认为，管理人员的亲和动机过强会影响其管理绩效，因为这会导致管理人员主观上过分关心下属，有时很难做出令别人不快却很必要的决策。另外，他还发现，在更为集权主义导向的国家中，员工大都是高权力动机和低亲和动机的。研究者们推测，动机的这种模式可能与滥用权力和独裁主义的管理风格有关。

每个人都在一定程度上具有上述三种动机，只不过在不同情境下它们所占的比重和表现程度不同罢了，而这主要取决于个人所处的职位。研究表明，强烈的成就动机和权力动机对于基层和中层管理者而言是一种可贵的优点，强烈的权力动机对于高层管理者来说尤为重要。一项研究发现，权力动机相对比较强烈的美国总统在任期间往往政绩出色。对管理者而言，强烈的亲和动机并不总是可取的，因为它会诱导他们花太多的精力去取悦他人（包括下属），而不是去做应该做的事情，来确保组织目标的实现。

（四）工作动机

工作动机（job motivation）指的是一系列能够激发与工作绩效相关行为，并决定这些行为的方式、方向、强度和持续时间的内部与外部力量。工作动机是一种看不见的、内在的、假设的结构，是工作激励的心理基础，长期以来受到西方组织行为学研究者的重视。

早期的工作动机理论认为，人们工作无非是为了收入、报酬等一些物质利益，后来的研究发现，物质利益不是工作动机中最重要的因素。20 世纪 50 年代曾有一项对美国员工进行的抽样调查，询问他们若是继承了一笔遗产，不用工作就可以生活得很舒服，他们是否愿意放弃工作，80%的人表示他们愿意继续工作。由此可见，物质需要并不是工作动机

的唯一来源。随后的研究表明，寻求归属感、满足自尊、追求自我实现等也是工作动机的来源。

国内学者对中国企业员工的工作动机进行了大量的调查研究，并取得了一系列的成果。张炼对企事业人员工作动机进行测量，结果表明：内在动机是主要的工作动机，但在工作动机的不同维度上，随着职业内容、单位性质不同而有所不同。张剑、郭德俊对中国企业员工工作动机的结构进行了研究，结果表明员工工作动机的结构包括追求胜任取向、外在报酬取向、他人评价取向、自我决定取向与良好关系取向五个因素，而且男性员工在追求胜任与自我决定取向上的得分明显高于女性员工，学历较高的员工在追求胜任取向上的得分显著高于学历较低的员工，但在外在报酬取向上的得分显著低于学历较低的员工。

总之，工作动机是一种复杂且重要的社会性动机，它是人们不辞辛苦勤奋工作的强大动力。在现代管理中，要做好激励工作，关键的一点就是激发员工的工作动机。因此，管理者应对员工工作动机的特点和变化规律有更深入透彻的了解和把握。

管理小贴士：

HR 经理：善用“动机”，你才会更优秀

锻炼和发展权力动机，培养适当的亲和动机与成就动机，你更有希望成为优秀的 HR 经理。

美国著名行为心理学家 David Clarence McClelland（1938—1998）通过二十多年的研究，提出动机需求学说，指出人类的三种需求动机分别是成就动机、权力动机，以及亲和动机。大部分人拥有并展示了三种动机的组合，它们的混合体决定并表现为一个员工或者经理人行为特征和工作风格，三种动机中占主导地位的动机对个人行为和风格的影响最大。

HR 经理人的“动机”调查

最近，国内著名管理咨询顾问张伟俊先生主持了“HR 职业生涯发展”的讨论会，与会 200 多名 HR 共同参与了一个简单的需求动机的调查，按照个人工作时间与精力分配多少，对代表不同动机的三个选项进行排序：

A. 把事情做得更完美/进一步提高工作效率

B. 建立良好的人际关系/受大家欢迎

C. 对他人施加更多、更有力的影响

结果显示，51%的 HR 将 A 项置于第一位，表现出以成就动机为主导动机；有 33%的人选择 C 项，表现出以权力动机为主导动机，另有 16%的 HR 将 B 项置于首位，以亲和动机为主导。

成就动机造就主力“干将”

调查发现，选择成就动机为导向的人中的大部分是企业人力资源部的主力“干将”，很多是优秀的 HR 助理、HR 专员，他们追求个人高成就，比他人做得更多，做得更好。其中一小部分目前处于 HR 经理以上的职位。但是当他们需要用自己的观点和做法去感染直线

经理，并取得最终的成效时，他们常常觉得力不从心，容易与直线经理产生摩擦。

根据 McClelland 的研究，成就导向的人，习惯性地思考如何将事情做得更好。一旦下决心做某件事情，一般都能取得进展，在公司里获得较多的表扬，提升机会也接踵而来。一个拥有很多这种类型员工的公司，较同类公司而言，发展会更快，利润也更高。但是，当这类人被提升到管理岗位时，工作效率就会下降。因为他们仍与原先一样竭力完成工作，并且期望他人也跟他们做的一样好，这就会导致与平级之间的摩擦，并使下属员工产生挫折感，无法发挥最大潜能。

亲和动机使人更受欢迎

具有亲和动机的 HR 期待良好的关系，希望到处受欢迎。他们是很好的团队合作者，追求的不是工作为导向的成就，而是与人愉快相处。当团队气氛良好，可以获得较多的人际接触和回馈时，他们的工作效率一般都非常高。然而，当他们被提升到领导者的位置，需要管理下属，推动各个不同部门的人力资源工作时，他们会遇到或强或弱的反对力量，内心感到痛苦。他们一如往常地关注其他人的感受，不希望造成冲突，为了维持原有的和谐，在妥协中失去作为一个领导者应有的公正性和判断力，导致工作无法推动执行。

三国时代的诸葛亮，他的动机类型是成就动机为主导，亲和动机极高，权力动机很低。他过分强调了个人的能力，没有培养下属，处理人际关系时又过于注重他人与自己的亲善关系，没能充分发挥团队的创造力和主动性，终于错过了蜀国称雄的大好时机。

权力动机更能影响他人

对一些比较成功的 HR 经理或者总监的访谈发现，他们大多选择了以权力动机为主导，并且拥有较高的成就动机和亲和动机。这些 HR 经理人比较倾向于追求权威和权力，能够主动运用人际影响力来与上级进行沟通并与企业的高层结盟，获得足够的权威以推动自己的工作。工作中，他们能制造出一个比较确定的工作道德规范和培养员工对组织的忠诚度。这主要是因为权力动机为主导的人为权力所驱动，具有一种去影响他人、给人以深刻印象以及造成冲击的需求。

现在很多企业的 HR 提倡要做企业的战略伙伴。要上升到如此高度，首先要做的就是让企业高层认识到人力资源的重要性，相信 HR 对于企业战略发展起到关键作用。张伟俊先生认为，HR 能不能参与到战略中来，关键要看 HR 能不能用专业的知识去影响“最终做决定的人”，权力动机高的 HR 通常在这点上占有很大的优势。

改变动机，成就成功

把 HR 按照行政职能来分成六级，前三级，包括人力资源助理、人力资源专员和人力资源主任（不考虑晋升的因素），最佳的动机曲线是成就动机最高，权力动机和亲和动机随后。后三级，包括人力资源经理，人力资源总监和人力资源副总裁，这些需要与企业高层合作来推动发展的经理人，他们的最佳动机曲线是权力动机最高，成就动机居次，亲和动机最后，权力动机和成就动机的要求都相对较高。

这里要说明的一点是，动机有别于能力。例如，成就动机非常高的优秀销售人员，他们可以表现出极高的亲和能力，但是他们不一定具备非常高的亲和动机，他们只是在成就动机驱动下，为了获得成就而运用他们的亲和能力作为一种手段。一旦达成销售目的，他

们就不再表现得那样的亲和了。所以，我们通常所说的 HR 最好要有亲和力，不是指亲和动机，而是指亲和能力。非亲和动机主导的 HR 甚至有可能比亲和动机主导的 HR 具有更强的亲和能力。

想要成为优秀 HR 经理人，却发现自己的主导动机是亲和动机或者成就动机，是否还有机会改变？McClelland 及其合作伙伴 David Burnham 的专项研究表明，动机是可以改变的。他们在两年中培训了来自不同公司的一千多位中高层经理。结果发现，至少三分之二的培训参加者发生了变化，更重要的是，他们的商业成果和员工的士气也得到了极大的提高。Burnham 幽默地说，“要教会一只年纪大的狗新的把戏不容易，教会领导者怎么去做却还是可以的。”

因此，期望成为优秀经理人的 HR，要通过培训和工作实践来锻炼和发展自己的权力动机，并且培养适当的亲和动机与成就动机，这样才能够为最终的成功做好准备。

（资料来源：中国人力资源开发网）

第四节　动机的激发及调适

激发和调适组织成员的工作动机，维持其积极的工作行为，是管理者的重要任务之一。本节首先从动机过程的内在起因、外部诱因及归因等几个方面探讨动机激发问题，之后再对个体动机的调适问题进行分析。

一、动机的激发

（一）激发动机的内在起因

心理学研究表明，人的行为源于个体的内在需要和内部唤醒状态，因此激发动机的首要策略就是从动机的内在起因入手。这在管理活动中具体表现为：

首先，管理人员要充分了解并激发组织成员的主导动机。个体在某一时期的社会活动中总会有一种主导动机，而它是个体行为的主要推动因素。

其次，由于动机的激发与需要的满足是密不可分的，所以管理者应及时满足员工的合理需要。

最后，使员工认清自己行为的目标。只有个体的内在需要与外在诱因（目标）相联系时，具有一定能量和方向性的驱力（动机）才会产生。

（二）设置与运用动机的外在诱因

外在诱因主要是指个体的行为目标和奖赏、惩罚等。众所周知，人是具有主观能动性的，一般说来，个体在付出行动之前就已经对行为的目标及其意义做出了自己的主观判断和心理预期。这样，行为目标（外在诱因）就可以转化为个体工作的内在动力，从而具有了动机作用。因而，目标设置就成为激发员工动机的重要方法，管理者应发挥目标的激励、

引导作用，为员工设置具有较强吸引力、难度适宜的目标，调动员工努力工作的愿望和热忱。此外，对员工行为结果的奖惩也会影响其工作动机。所以，激发动机也可从行为入手，作为管理者，要勤于分析员工的行为及其意图，及时对员工的行为进行合理的强化（奖赏和惩罚），奖惩分明，以此来提高员工工作动机的水平和强度。

（三）利用员工的自我调节能力

在动机的激发中，自我调节发挥着重要的作用。目标、强化这些外在因素也只有通过自我调节才能转化为个体的内在动力。具体到管理实际中，帮助员工提高自我调节能力可以从以下几方面入手：

首先，管理人员要使员工正确认识自己，对自己的工作保持适度的期望值，对自己的期望既不能太高，远远超出自己的能力所及，也不能过低，使工作失去了挑战性与吸引力。

其次，要帮助个体建立起较高的自我效能感，使个体在工作中不断地获得成功体验，对自己的能力产生一种较强的自信心。

第三，要锻炼员工的意志力，提高其意志水平。当员工自身具备较强的意志力时，他就会主动去克服困难、寻求工作中的挑战，行为也会更有坚持性。

最后，管理人员要及时向员工反馈其工作成效，利用员工自身的反省意识和能力，帮助其提高工作能力，这样，员工在工作能力上的不断进步与提升就会进一步激发员工工作的愿望和动机。有关研究证实，对个体行为结果的及时反馈能够激发其活动的动机，进而产生良好的激励作用。

（四）对员工进行归因训练

由于个体对行为结果的归因直接影响到其随后的动机水平，所以对个体进行归因训练就成为一种激发动机的有效策略。所谓归因训练是指一套训练程序，训练者运用说服、讨论、示范、强化等措施对个体进行有目的、有计划的指导，其目标是通过系统的干预，使个体原本不利于提高后继行为动机的归因模式转化为有利于提高动机水平的模式。例如，某员工在一次职务晋升中失败了，他将自己的落选归因于自身能力的不足，归因于自己不被领导重视和认可，从此一蹶不振，在工作中表现消极，总带着一种抑郁、悲观失望的情绪，失去了努力拼搏的动力。此时，管理者就可以从改变员工的归因模式着手，引导其作出外部的、不稳定的、可控制的归因，如公司该职位的名额有限、竞聘时优先考虑经验丰富的人员等，使其改变原先不合理的归因模式，正确归因进而保持努力工作的热情。总之，个体对事件结果归因的正确与否极大地影响着其随后的行为动机，在工作中，领导者要有意识地引导下属进行积极的归因，激发起员工工作的动机，保证员工工作的积极性和主动性。

二、动机的调适

以上探讨了动机的激发问题，事实上，在现实的组织情境中，员工的动机不仅需要激发，还需要适时适度地进行调适。这样才能使动机始终维持在适当的水平，成为员工行为

积极性的持久源泉和动力。下面从动机方向、动机强度、动机稳定性和动机清晰度等四个方面来分别探讨如何合理有效地调适动机。

（一）动机方向的调适

动机的方向主要反映动机的思想倾向。不同个体做相同事情的动机方向有可能不同。企业中，同样是努力工作，有的员工是为了实现自我的价值，而有的单纯是为了追求权力和物质报酬。后一种动机虽然也可以改善工作绩效，但从长远角度看，对于组织发展和个人成长都会带来不利影响。因此，对于方向上有失偏颇的动机，管理者要进行适度的引导和调适，通过与员工共同构建企业发展的愿景，将员工的动机引导到与组织目标一致的方向上。

（二）动机强度的调适

依据前述的耶克斯—多德森定律，应根据任务的难易程度来调整动机的强弱。对简单机械的重复性劳动而言（如装配流水线作业），较高的情绪兴奋水平和动机状态有利于做出好成绩，因此，需要尽可能提升员工的唤醒水平，有效利用外在奖励充分调动员工的动机；对于中等难度的工作任务，尤其是需要与人配合的工作，中等程度的动机水平更有利；而对于需要创新思维的复杂任务，适当减少外在压力，降低动机水平更有利于任务的完成。

（三）动机稳定性的调适

即便个体有方向正确且强度适当的动机，但如果这种动机状态不稳定，难以维持，也不能保证任务的顺利完成和目标的顺利实现。因此，对动机的稳定性进行监控和调适，采取相应策略保持个体适当的动机状态，也是动机管理的重要任务之一。关于组织环境中如何做到这一点，将在下两章有关激励的内容中加以介绍和分析。

（四）动机清晰度的调适

动机的清晰度是指个体对自身行为动机的意识程度，它是动机的一个重要特性。一般说来，只有当个体的行为动机被自己清晰地意识到时，才会对行为产生主动的影响作用。否则，即便内心里有某种动机，但自己没有意识到或意识不够明确，就会使行为的方向、强度和稳定性大打折扣。人们不仅会因为具有某种动机而行动，还会因为对这种动机的意识程度不同而表现出不同的行为强度和稳定性。因此，采取适当的措施，对动机的清晰度进行调适，也是动机管理的重要内容之一。

本章小结：

1．动机是一种由目标或对象引导、激发和维持个体活动的内在心理过程或内部动力。它包括以下三个既相互作用又相互独立的因素：需要、驱力和诱因。动机主要有激发、导向、维持和调整等功能。

2．动机和行为之间有着复杂的关系，具体表现为：有动机不一定有行为，有行为不一定有动机；同一动机可能产生几种不同的行为，同一行为可能受不同动机驱使。动机强度

影响工作效率，即动机不足或过强，都会降低工作效率，只有动机处于最佳水平，工作效率才会提高。

3．心理学中的动机理论按出现时期不同分为两大类：生物因素的动机理论和认知因素的动机理论。前者主要包括本能理论、驱力理论和诱因理论等，后者主要包括动机的认知一致性理论和动机的归因理论等。

4．按照不同的标准可把动机分为不同的种类，如生理性动机与社会性动机、外部动机与内部动机、主导动机与辅助动机。其中社会性动机主要有：成就动机、权力动机、亲和动机和工作动机等。

5．激发动机的主要策略有：激发动机的内在起因、设置与运用动机的外在诱因、利用员工的自我调节能力以及对个体进行归因训练等。在动机调适方面，主要包括动机方向的调适、动机强度的调适、动机稳定性的调适和动机清晰度的调适等。

本章思考题：

1．分析动机的概念和不同类型动机的含义。
2．试述动机理论的发展历程。
3．简述耶克斯—多德逊定律及其管理启示。
4．简述认知失调理论，分析其在组织管理中的应用价值。
5．简述归因理论，分析其在管理活动中的应用价值。
6．激发动机的主要策略有哪些？结合管理实践谈谈你的看法。
7．动机调适的基本内容有哪些？试结合实际对某一内容的调试进行分析。

第五章　激励理论

现代组织管理的核心是对人的管理。任何组织的创建和发展，组织内的一切物流、资金流、信息流的运作都是由人来完成的，人是决定组织成败的最关键因素。组织中人的积极性的高低直接影响工作的绩效，而提高人的工作积极性，就离不开激励。因此，激发、调动组织成员的工作积极性既是企业管理的基本内容，又是管理心理学研究的核心问题之一。本章首先对西方的激励理论加以介绍和分析，然后介绍和分析我国的激励理论，最后对中西方的激励理论做一简单的比较，指出激励理论的文化限制和文化相对性。

第一节　西方的激励理论

什么是激励？美国管理学家贝雷尔森和斯坦尼尔（Berelson & Steiner）给激励作了如下的定义："一切内心要争取的条件、希望、愿望、动力等都构成了对人的激励……它是人类活动的一种内心状态。"由于人的行为由动机而引发，因此，研究者认为，激励是指激发人的内在动机，鼓励人朝着所期望的目标采取行动的过程。第四章介绍的动机理论主要说明了人采取某种行为的原因，而对于管理者来说，关注的焦点则是怎样使员工按照组织期望的方向产生特定的行为。

激励理论是关于激励的基本规律、机制及方法的概括和总结，是激励在管理活动中赖以发挥功能的理论基础。自 20 世纪 20 年代以来，西方的许多管理学和心理学家分别从不同角度进行研究、探索，提出了许多激励理论。这些理论从不同的侧面研究人的行为动因，但每一种理论都具有其局限性，不可能用一种理论去解释所有行为激励问题。各种激励理论可以相互补充，从而使激励理论不断完善。

无论西方的激励理论多么千差万别，但它们都是围绕以下三个问题展开的，都是为了回答这三个问题：（1）是什么激发或驱动行为的？（2）是什么引导行为方向的？（3）是什么使行为得到维持的？下面就以这三个问题为主线，依据不同理论回答问题的侧重点不同，对西方的激励理论加以介绍和分析。

一、是什么激发或驱动行为的

在这个问题上，研究者们已经提出了多种观点，下面仅就两种影响较大的理论加以分析：一种是马斯洛（A.H.Maslow）的需要层次论，另一种是赫茨伯格（F.Herzberg）的双因素论。

（一）需要层次论

需要层次理论是由美国心理学家马斯洛在 1943 年所著的《人的行为动机》一书中提出

来的，该理论在世界上广为流传，已成为应用最普遍、最主要的激励理论之一。

1．需要层次论的基本内容

马斯洛的需要层次论主要有三个观点。

（1）人的多种需要分为五个层次

马斯洛认为，需要是人类行为的积极的动因或源泉。需要引起动机，动机驱动行为。因此，弄清楚了人类的基本需求结构或层次，就能很好地说明、解释、预测和控制人类的行为。他按照由低级到高级的顺序，将人的需要分为五个层次：生理需要、安全需要、社交需要、尊重需要与自我实现需要。

生理需要是人类维持其生命和生存最基本的需要，也是需要层次的基础，包括对食物、水、性以及避免寒冷或炎热的需要等。对于缺乏这类东西的人来说，其主要的行为动机受生理需要支配。例如，一个极端饥饿的人，将会把全部的精力用在觅食活动上，对他来说有食物的地方是最理想的去处。马斯洛曾说："一个人如果同时缺少食物、安全、爱情和被尊重等，则食物的渴求可能最为强烈。"生理需要是人类最低层次的需要。一旦这种需要被满足，人们便开始追求更高层次的需要。

安全需要是指对避免危险、威胁和剥夺的需要。最常见的安全需要之一，就是保护个人免受肉体上的危险，如火灾或事故等。在工业部门，常见到这类标牌："此区禁止吸烟"、"超过此处必须佩戴护目镜"等，也是管理者为满足安全需要所做的一种努力。第二种安全需要是经济保障。员工的各种福利待遇，如事故保险、健康保险以及人寿保险等，均有助于满足这类需要。第三种安全需要是对有秩序的、可预知环境的需要。安全需要的涵义很广泛，从世界和平、社会安定直至个人的安全等都属于安全需要的范畴。安全需要是人类安居乐业的基本保证。

社交需要是指个人对爱、情感和归属的需要，主要包含两方面内容：一是爱的需要，即人都希望伙伴之间、同事之间关系融洽，保持友谊和忠诚，希望得到甜美而忠贞的爱情，希望爱别人，也渴望得到别人的爱。二是归属的需要，即个体都有一种要求归属于一个集团或群体的感情，希望成为其中的一员并得到相互关心和照顾。组织管理者应该认识到，当社交需要成为主要的激励来源时，员工们会把工作视为寻求建立温馨、友善、和谐人际关系的机会。

尊重需要是指个体希望获得成就感以及得到他人对自身价值的承认与尊重。尊重需要分为两类：一类是希望有实力、有成就、能胜任、有信心以及要求独立和自由；另一类是要求有名誉或威望，受到别人的赏识、关心、重视或高度评价。马斯洛认为，尊重需要得到满足，能使人对自己充满信心，对社会满腔热情，体会到自己生活在世界上的用处和价值。但尊重需要一旦受到挫折，就会使人产生自卑感、软弱感、无能感，从而使人失去生活的基本信心。关注员工尊重需要的管理者一般会通过公开奖励以承认其工作业绩等方式激励员工。这类管理者会采用在公司报刊上撰写文章或在公报栏里发布消息等方式对员工的良好表现进行表扬。当然，也有部分员工不喜欢公开表扬而宁愿私下里被肯定成绩。

自我实现需要是人的最高层次的精神需要。它是指发挥自我内在潜力，实现自己理想和抱负的需要。产生这种需要的人决心发挥自己最大的能力完成难度较大的工作任务，成就一番事业，努力使自己成为理想的人。这种需要往往通过胜任感和成就感获得满足，意

乐家创作出天籁之音，作家写出千古流芳的作品，诗人吟唱出脍炙人口的诗篇……这些都是借以达到自我实现的途径。注重自我实现需要的管理者会鼓励员工们进行工作设计，发挥一技之长，或者给予班组自由安排工作任务的权力。

（2）人的需要分等分层，像阶梯一样逐渐上升

马斯洛认为，人的五个层次的需要是由低向高排列的。一般来说，只有在低层次的需要得到满足之后，人才会进一步追求较高层次的需要，而且低层次需要满足的程度越高，对高层次需要的追求就越强烈。马斯洛将五种需要划分为高、低两级。生理需要和安全需要属于低级需要，这些需要通过外部条件使人满足，如借助于工资收入满足生理需要，借助于法律制度满足安全需要等。社交需要、尊重需要和自我实现需要是高级需要，这些需要是从内部使人得到满足的，而且人们对这些高级需要永远不会感到完全满足，因此通过满足员工的高级需要来调动其工作积极性，具有更稳定、更持久的力量。人在不同的发展阶段，其需求结构也不同。一般来说，在同一时期内，人们可以同时存在几种需要，但总有一种需要占支配地位，它决定着人们行为的方向。

（3）未被满足的需要才有激励作用

需要是促使人产生某种动机的内在基础。当一个人一无所求时，也就没有什么动力与活力；反之，若一个人有所需求，就必然存在着激励的因素。五个层次的需要是人生来就有的，但每一个人的需求强度、显露程度可能不同，这由生物的、文化的和环境的因素共同决定。另外，当某种需要得到满足之后，这种需求也就失去了对行为的唤起作用，下一层次的尚未满足的需求就会成为人们行为的动力。高层次的需要，不仅内容比低层次需要更广泛，实现的难度也更大。马斯洛 1943 年曾经指出，85%的生理需要和 70%的安全需要一般会得到满足，但只有 50%的社交需要、40%的尊重需要和 10%的自我实现需要能得到满足。

2．需要层次论在企业管理中的应用

马斯洛的需要层次论第一次系统地阐述了人的需要与行为之间的关系，在现代企业管理中具有广泛的应用价值。表 5.1 列出了不同层次需要对应的追求目标及相应的管理措施。

表 5.1　需要层次论与管理措施相关表[①]

需要的层次	诱因（追求的目标）	管理制度与措施
1．生理需要	薪水，健康的工作环境，各种福利	身体保健（医疗设备）、工作时间（休息）、住宅设施、福利设备
2．安全需要	职位的保障，意外的防止	雇佣保证、退休金制度、健康保险制度、意外保险制度
3．社交需要	友谊（良好的人群关系），团体的接纳与组织的一致	协谈制度、利润分配制度、团体活动制度、互助金制度、娱乐制度、教育训练制度
4．尊重需要	地位、名誉、权力、责任、与他人薪水之相对高低	人事考核制度、晋升制度、表彰制度、奖金制度、选拔进修制度、委员会参与制度
5．自我实现需要	能发展个人特长的组织环境，具有挑战性的工作	决策参与制度、提案制度、研究发展计划、劳资会议

[①] 转引自俞文钊．管理心理学．上海：东方出版中心，2002：第 249 页

从表 5.1 可以看出，管理者可以根据五种需要层次对员工的多种需要进行归类和确认，然后了解员工未满足的或正在追求的需要是什么，采取相应的管理制度与措施进行激励，既不能落后、又不能超前于员工的需求状况。这就是说，当员工的低层次需要未得到满足时，不要奢望他们有更高的献身精神，应把解决他们的衣食住行问题放在首位；而当员工的低层次的需要被满足了时，又不能一味地停留于用物质奖励的办法来刺激他们，应该创设更好的工作和人际环境，满足他们更高层次的需要。在某种程度上说，一个企业或组织员工需求层次的高低，应作为该企业或组织管理成败的重要标志之一。员工的需求层次愈高，说明这个企业的管理愈成功；反之亦然。

3．对需要层次论的评价

西方管理心理学对马斯洛的需要层次论的评价是各不相同的。总的来说，既重视这一理论，又指出了这一理论的不足之处。

一方面，马斯洛的需要层次论有其科学性的一面。第一，马斯洛对人类的需要进行系统研究，把千差万别的人类需要归纳为五个层次，并就五个层次需要的内容和层次间的关系作了详细阐述，这与以往心理学所进行的零散、不系统的研究相比，无疑是前进了一大步。因此，需要层次论对心理学的需要动机理论发展是一个重要的贡献。第二，马斯洛提出的需要层次性和需要由低级向高级发展的趋向，在某种程度上是符合人类需要发展的一般规律的。一个人从出生到成年，其需要的发展基本上是按照马斯洛提出的需要层次进行的。第三，马斯洛的需要理论为企业管理指出了调动积极性的工作方向和内容。例如，任何企业都应该从物质和精神两方面去满足员工的合理需要，同时要根据不同员工不同的需要层次，针对性地采取不同的管理措施。

另一方面，马斯洛的需要层次论也有局限性和不足之处。第一，马斯洛的需要理论绝大部分谈的是人的自然需要，忽视了社会存在对人的成长所具有的影响。马斯洛认为人的需要都是本能的活动，都是与生俱来的。人有生理需要是为了维持自己的生存；人有安全需要是出于趋利避害的本能；人有社交需要是为了享受生活的乐趣；人有尊重和自我实现的需要是为了能出人头地，高高在上。这样就完全否定了社会存在对人的成长的决定性影响，宣扬个人主义的实质非常明显。第二，马斯洛的需要层次论带有一定的机械主义色彩。把人的需要层次看成是固定的程序，认为只有满足了低一级的需要层次之后，才能进入下一层次的需要，这显然是机械的。它忽视了人的主观能动性，忽视了高层次需要对低层次需要影响的一面，忽视了为了崇高理想，人可以忍受物质生活困难带来的痛苦，甚至牺牲宝贵生命的事实。

（二）双因素论

双因素理论是美国心理学家弗里德里克·赫茨伯格提出的。像马斯洛一样，赫茨伯格试图弄清楚究竟是哪些因素激发或驱动组织中员工的行为，但他不是在个人的内在需要中去找原因，而是将目光投向了个人之外的工作条件和环境，试图直接查明哪些因素影响员工的满意程度及其对工作的积极或消极态度。

1．双因素论的基本内容

20 世纪 50 年代后期，赫茨伯格在匹兹堡心理研究所与其同事们，对该地区的 11 个工

商业机构中的 200 位工程师和会计师进行了工作满意感方面的调查，发现使受访人员不满意的因素多与他们的工作环境有关，而使他们感到满意的因素通常是由工作本身所产生的。根据调查结果，赫茨伯格提出了双因素理论，把企业中影响人的积极性的因素，分为激励因素和保健因素两大类。

激励因素（motivative factors）是导致满意的因素，往往与工作内容或工作成果有关。这类因素有：成就、赏识、工作本身的吸引力、责任、发展等，如表 5.2 所示。它们能产生直接的激励作用，使员工对工作有一种满足感。这是影响人们工作的内在因素，能够促进人们积极进取，不断提高工作效率，因此将这类因素称为激励因素。

保健因素（hygiene factors）是导致人们不满意的因素，往往与工作环境或外在因素有关。这类因素有：公司政策与行政管理、监督、工作条件、人际关系、地位、安全和生活条件以及薪金等，如表 5.2 所示。保健因素没有激励人的作用，但却带有预防性，具有使人保持积极性、维持工作现状的作用。打个比方：你在回家的路上滑了一跤，手部摔破了皮。你回家后在伤处涂上碘酒。两周后你的手完全康复。虽然碘酒没有使你的手比受伤前更健康，但却防止了伤口进一步恶化，而且帮助你的手恢复原状。这就是保健的作用。

表 5.2　激励因素和保健因素

激励因素（工作本身）	保健因素（环境）
成就	公司政策与行政管理
赏识	监督
工作本身的吸引力	工作条件
责任	人际关系
发展	地位
	安全和生活条件
	薪金

关于激励因素与保健因素与人们的工作满意度和工作态度的关系，赫茨伯格指出，传统上认为满意的对立面是不满意的看法是不正确的。实际上，有两个独立的满意程度量表：一个是反映从不满意到没有不满意变化的量表，主要受保健因素的影响；另一个是反映从没有满意到满意变化的量表，主要受激励因素的影响。在这两个分量表上，人们对工作的态度的变化也是不同的，如图 5.1 所示。



图 5.1　赫茨伯格的双因素理论

2．双因素理论对管理的启示

很显然，按照双因素理论，在管理实践中，为了增加员工的满意程度，充分调动他们的工作积极性，就不能仅仅限于改善企业的物质条件或工作环境这些保健因素，更为重要的是改善激励因素，为每个员工提供发挥才能的机会，增强他们的成就感和责任心，让他们感到有前途、有奔头。否则的话，即便作业的工作环境和条件再好，付出再多，也只能让员工感到没有什么不满意的，而不能让他们感到满意，因而难以起到调动员工工作积极性的作用。许多管理者已经从双因素理论中得到启示，在企业管理中，采用工作丰富化、工作扩大化和弹性工时等具体措施来调动员工的工作积极性。关于这方面的具体内容将在第六章中具体介绍。

需要指出的是，双因素理论是在美国的文化背景下提出来的。我国与国外的社会制度、国情、民族传统不同，因而在企业中，哪些因素应该属于保健因素，哪些因素应该属于激励因素，与国外的划分是有差异的。对于每一个人来说，激励因素和保健因素也各不相同，同一个因素，对一个人来说是激励因素，而对另一个人来说可能属于保健因素。因此实际应用时，应区别对待不同人的保健因素和激励因素，才能提高激励效果。

二、是什么引导行为方向的

个体的行为不仅由内在或外在力量所驱动，而且有一定的目标和方向。处于某个组织或机构中的个人，当被某种力量激发或驱动采取行动时，通常不会不计后果，莽撞行事，他总是要将其行为指向特定的目标。有些人可能将其行为指向较高层次的工作目标，而另一些人可能将其行为指向工作之外的目标。因此，弄清楚是什么在引导个人的行为方向，对于管理工作也很有启发意义。在这个问题上有两种理论影响较大：一是洛克（E.A.Locke）等人提出的目标设置理论（the goal-setting theory），另一种是弗鲁姆（V.H.Vroom）提出的期望理论（the expectancy theory）。

（一）目标设置理论

人的行为大多是有目的性的行为。一个人的行为有无目的性，其结果大不一样。心理学的许多研究也表明，漫不经心的练习是没有什么作用的，在掌握技能的过程中，练习者为自己确立一定的目标，对于提高练习的效果具有重要意义。例如，练习游泳时，练习者确定当天或一个阶段内所要达到的目标，并且向这一目标奋斗，就会有助于练习效果的提高。正是基于这样的观点，洛克等人提出了目标设置理论。

1．目标设置理论的基本内容

目标设置理论的主要观点是：管理者不需要直接控制员工的行为，即使这样做了，也常常是事倍功半，效果也不会好；相反，通过设定一些为员工所接受和认同的目标来指引其行为，常常可以起到事半功倍的效果，因为目标本身具有内在激励价值。同时，管理者还可以通过对员工提供目标达成程度的回馈信息，来使目标成为一个更加有效的激励因子。洛克等人设计了一个个体目标设置与绩效模型，如图 5.2 所示。

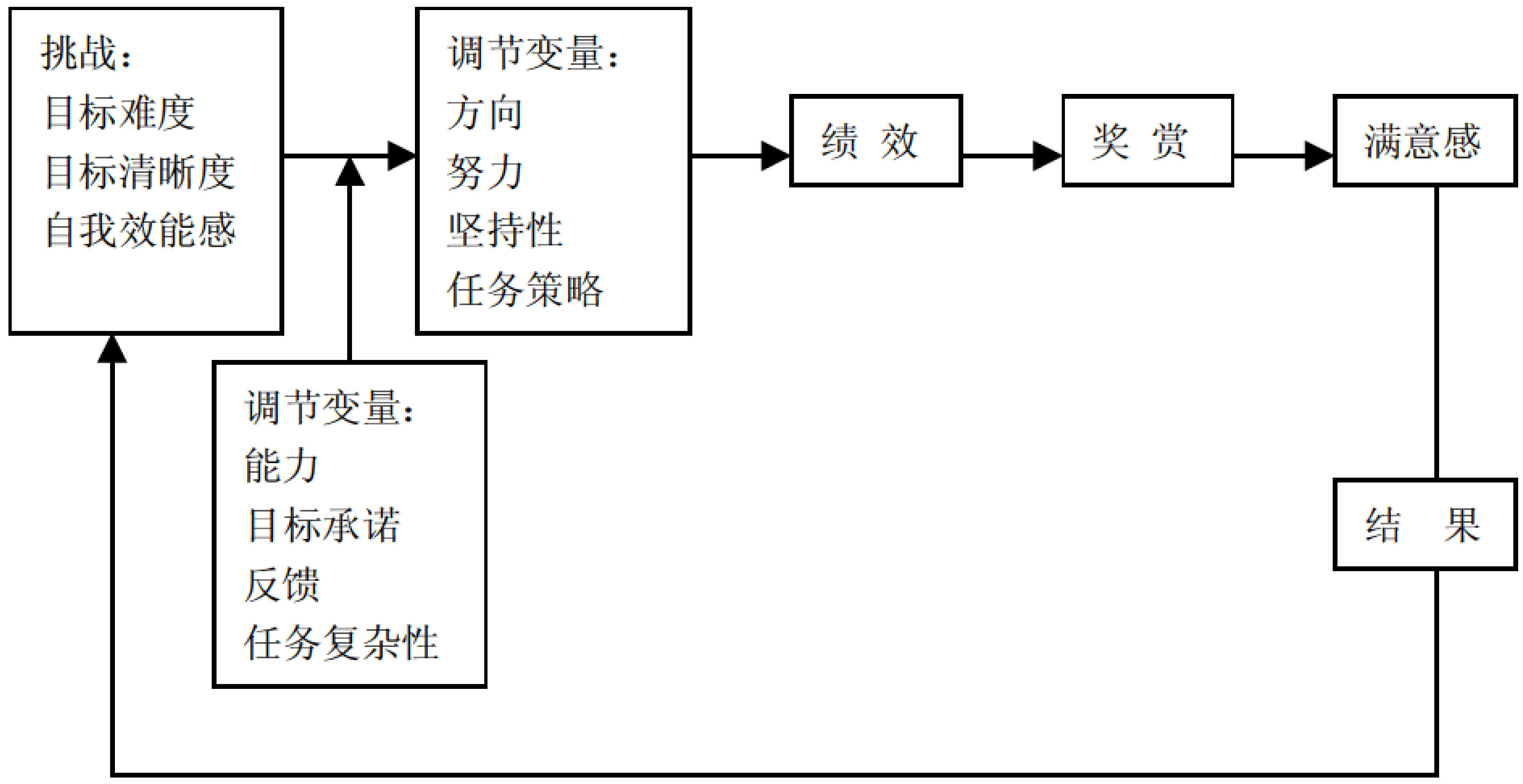


图 5.2 目标设置模型

从这个模型中可以看出引发个体高绩效水平的因素及其关系。该模型的基本观点是把目标看作一种激励因素，因为它可以让人们将目前的绩效与期望达到的目标进行比较。从某种程度上来说，人们一般会认为，如果他们目前的水平达不到目标的要求，他们就不会感到满足。但只要他们相信，经过努力是可以达到目标的，他们就会努力工作并实现目标。

目标有两个关键特征：目标难度和目标清晰度。

目标难度是指一项目标的挑战性和需要努力的程度。目标难度会影响员工对目标的接受与认可。当员工认为一项目标可行且有意义时，他们就愿意付出努力追求该目标；而当一项目标太简单或太困难时，目标不会被接收和认可，因此也就失去了激励的作用。

目标清晰度是指一项目标的明确与清楚程度。许多事实证明，把目标订得明确要比笼统地要求“你们好好干吧”好得多。国外研究表明：若干人参加试验，先根据他们的成绩、能力和态度对其评分，然后将他们分别派到一个积极性高和一个积极性低的小组去。给那些积极性低的小组布置十分具体的工作目标，而对积极性高的小组则只告诉他们“好好干”。结果表明，那个积极性低的小组，无论在工作成绩还是对工作态度方面，都很快赶上并超过了高积极性的小组。

2．目标设置理论在管理中的应用

根据上述目标设置理论，在实际的管理活动中，管理者应注意以下两点：首先，应及时了解员工的目标设置状况。如果管理者了解员工所拥有的目标，知道他们重视什么或期望得到什么，就有了制定激励措施的依据。其次，管理者应采取一定的措施引导员工的目标设置，目前在很多企业中盛行的目标管理就是从这一角度出发的。

目标管理（Management by Objectives），简称 MBO，是一种管理技术，其创始人是美国管理心理学家德鲁克（Peter.F.Druker）。目标管理是指管理者和员工为工作绩效和个人发展共同设置目标，定期对员工在实现目标过程中的进步进行评价，并对个体、团队、部门和组织的目标进行整合。

目标管理的具体做法可分为三个阶段：第一阶段是目标的设立。企业设立总目标，员工积极主动地参与目标设置，或者就选择什么目标提出建议，或者至少能和上级一起讨论并同意这些目标。目标要订得尽可能具体。每个部门根据总目标，会同上级订立部门目标，员工再根据所属部门的目标订立个人目标，从而形成一个目标链。第二阶段是过程管理。用一整套管理控制的方法去实施目标，但主要是放手让员工发挥各自的积极性，完成自己所订立的个人目标。具体实施办法可由各人自行确定，不必人人一样、千篇一律。第三个阶段是对照既定目标来考评效果，并讨论未达成目标的原因，同时为下一个目标管理周期创造更好的条件，以利于设置新目标。

目标管理就是强调围绕企业的经营或活动目标开展管理活动。在此，目标设置显得十分重要，它可以将员工的行为统一在一个方向上，调动员工的积极性，提高工作效率。

（二）期望理论

对激励问题解释得最全面的，是美国心理学家弗鲁姆于 1964 年在《工作与激励》一书中提出的期望理论，虽然它也受到一些批评，但有大量的研究支持这一理论。

1．期望理论的基本内容

该理论认为，只有当人们认为实现预定目标的可能性很大，并且实现这种目标又具有很重要的价值时，该目标才会对人产生最大的激励作用。换句话说，决定行为的因素有两个：期望值和效价。激励的程度由期望值与效价的乘积决定，用公式表示为：

激励程度（M）＝ 期望值（E）×效价（V）

其中，激励程度是指调动人的积极性、激发人内部潜力的强度。

期望值是指人们根据自己的经验判断一定的行为能够导致某种结果、满足需要的概率。例如，一个人认为自己考取大学的概率很高，相当有把握，那么他的期望值就很高；如果他认为自己考取大学的可能性很低，那么他的期望值就很低。

效价是指达到目标对于满足个人需要的价值。例如，有一个人希望通过努力得到升迁的机会，这就表明，他的升迁欲高，升迁在心目中的效价就很高；如果一个人对升迁漠不关心，升迁对于他来说，效价就很低。

该公式说明，假如一个人把目标的价值看得很高，估计实现目标的概率很大，那么该目标对他的激发动机就越强烈，焕起的内部力量也就越大。不妨举例来说明三者之间的关系：一位公司销售经理对他的一位销售员说，如果他今年能完成 1000 万元的销售额，公司将奖励他一套住房。这时组织的目标是 1000 万元的销售额，个人的目标是一套住房。如果一套住房正是这位销售员梦寐以求的，那么住房对他的效价就很高；而且，销售员根据今年的行情，认为自己再努力一点，1000 万元的销售额是可以完成的，那么实现目标的期望值也很高，这时这位销售员的工作积极性就会很高。

效价和期望值的不同结合，会产生不同的激励力量，其情况有以下几种：

（1）E 高 × V 高 ＝M 高

（2）E 中 × V 中 ＝M 中

（3）E 低 × V 低 ＝M 低

（4）E 高 × V 低 ＝M 低

（5）E 低 × V 高 ＝M 低

可见，要使被激励对象的内部力量达到最大，效价值和期望值都必须很高。当效价和期望值中有一项的值很低时，这件事对被激励对象来说就缺乏激励力量。

为了使激发力量达到最佳值，弗鲁姆提出了人的期望模式，如图 5.3 所示。

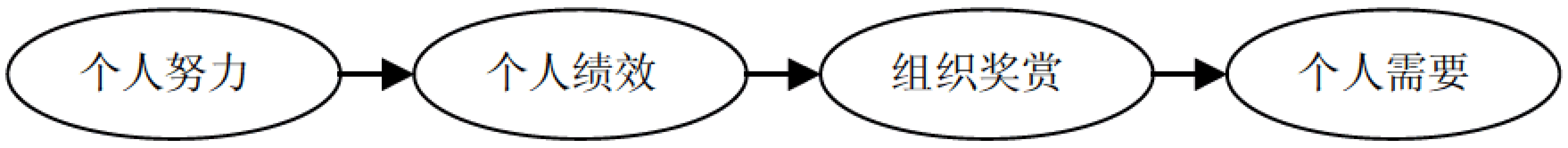


图 5.3 简化的期望模型

弗鲁姆认为，根据人的期望模型，为了有效地激发员工的工作动机，需要正确处理好以下三种关系：

第一，努力与绩效的关系

人们总是希望通过一定的努力达到预期的目标。如果个体主观上认为通过自己的努力达到预期目标的概率高，就会有信心，就可能激发出很强的工作动机；反之，就会失去内在的动机，导致工作不积极。

第二，绩效与奖励的关系

人们总是希望获得成绩后能够得到奖励。这种奖励是广义的，既包括提高工资、多发奖金等物质奖励，也包括表扬、自我成就感、被赏识等精神奖励。对于一项干得出色的工作，甚至连象征性的口头上的认可都可能奏效。正如马克 • 吐温（Mark Twin）所说：“我能仅凭一句赞美而生活两个月。”但是，高绩效是否一定能带来奖励呢？员工也会对此作出主观上的估计和判断，这种主观估计和判断的结果也会影响工作的热情和动力。如果员工认为好的绩效带来奖励的概率较高，就会增强工作动机；反之亦然。所以，组织能够看到员工的优异表现，并及时给以反馈、表示认可是十分重要的。基于这种思想，一些公司已采取了非常有创意的措施（见表 5.3）来及时地对员工的表现表示认可。

表 5.3 一些公司的非金钱认可方式

公　　司	商 业 性 质	认可的方式
肯戴尔	药品的临床测试设计	将所有员工（288 名）从事他们所喜爱的户外运动的照片沿着走廊张贴
洛哈德特电镀公司	钢铁制造厂	允许抛光部门的员工实行自我管理
辛辛那提机场附近的 50 家小型公司	多种经营	这些公司联合起来为住得较远的员工提供免费的交通服务
佩恩有限公司	环境咨询公司	建立电话交流设备以使某些员工在家中工作更方便

第三，奖励与满足个人需要的关系

奖励要能够满足个人的需要，如生理需要、尊重需要、自我实现需要等。由于人与人之间在年龄、性别、社会地位、经济条件等方面存在差别，反映在需要上也有明显的个体差异。因此对同一种奖励，不同的人所体验到的效价不同，所以同样的奖励对不同员工所具有的吸引力也不同。员工也会对得到的奖励满足自己需要的程度或可能性作出估计和评

价，这种估计与评价也会影响奖励的效果。

2．期望理论对管理的启示

期望理论的优点在于，它避免了马斯洛和赫茨伯格研究方法中的简单化的倾向，将员工工作动机的产生过程考虑得更加复杂，因而更加符合现实情景中员工动机变化的规律，为管理者进行员工激励提供了许多有价值的启示。

第一，管理者应设法确定每个员工所应得的报酬。具体办法是：观察员工对于不同报酬的反应，以及向员工们询问他们希望从工作中获取何种类型的报酬。

第二，管理者应该采用可观察、可计量的术语对员工表现好、一般和差的等级进行界定，使员工们明白公司对于他们的期望是什么。

第三，管理者应该确保为员工们所设置的业绩水平标准是他们经过努力能达到的。

第四，管理者应将企业特殊的业绩要求与员工所渴望的工作报酬直接挂钩，当员工实现或超额实现了业绩标准时，应及时兑现相关的奖励，给于赞扬或认可。例如，如果一位员工达到了升职所要求的业绩水平，那么他就应该被立即提升。

第五，管理者应确保结果或报酬的变化大到足以激发个体作出努力的程度，还应该保证所给予的奖励或回报是员工所渴望的、有吸引力的。微不足道的奖励或不适当的奖励对员工都是没有意义的，不具有激励价值。

三、是什么使行为得到维持的

当企业员工受到某种因素的驱动，产生了一种目标行为时，这种行为如何才能得到维持呢？关于这个问题也有两种较有影响的理论：一种是源于斯金纳（B.F.Skinner）的操作性条件反射概念的强化理论（reinforcement theory）；另一种是亚当斯（J.S.Adams）的公平理论（equity theory）。

（一）强化理论

本章前面所探讨的激励理论都是从纯理论的角度出发，这种方法最大的问题是不易进行科学的度量和观察。例如，如何度量当事人对地位或者自尊的需求？美国心理学家斯金纳率先提出了强化理论，试图解释行为及其后果之间的关系。

1．强化理论的基本观点

强化理论认为人的行为是由外部因素控制的，控制行为的因素称为强化物。强化物是在行为之后紧接着引入的一个结果，它提高了行为重复的可能性，因此，强化理论认为行为是其引入的结果的函数。按照这种理论，对员工的工作表现或成绩的强化（如物质奖励等）是使其行为得以维持的主要手段。斯金纳提出了几种行为改造策略，如图 5.4 所示。

	引入结果	没有结果	消除结果
行为增加或保持	正强化		负强化
行为减弱或消退	惩罚	消退	惩罚

图 5.4　行为改造策略

（1）正强化。正强化是指某种行为后，引入一种结果（这种结果通常来说是积极的，既包括奖金等物质奖励，还包括表扬、提升、改善工作关系等精神奖励），使该行为得到增加或保持。也就是说，奖励那些符合组织目标的行为，以便使这些行为得到进一步加强。员工在成功完成一项重大项目之后得到的奖励就是一种正强化，奖励的目的是为了使类似的行为得到维持或加强。

（2）负强化。负强化是指某种行为后，去除或避免一种结果（这种结果通常来说是消极的，如扣发奖金、批评等），使该行为得到增加或保持。员工上班迟到会受到上司的批评，一旦他意识到错误，按时上班了，上司就不再批评，这就是一种负强化。正强化和负强化都是为了使某种行为得到保持或增强。

（3）惩罚。惩罚就是对不良行为给予批评或处分，它可以减少这种不良行为的重复出现，弱化该行为。惩罚既可以是引入一种不愉快的结果（如要求加班、给予批评），也可以是取消一种愉快的结果（如扣发奖金、不再给予表扬）。

（4）消退。消退是指对某种行为不采取任何强化措施，既不奖励也不惩罚。这是一种消除不合理行为的策略，因为倘若一种行为得不到强化，那么这种行为的重复率就会下降。例如，当员工表现出的良好行为屡次都被上司所忽视时，那么由于这些良好行为没有得到及时强化，所以这些良好行为就会慢慢消失。惩罚和消退都是为了使某种行为减弱或消失。

此外，时间因素也会影响强化的效果。一般而言，强化时间的安排大致可分为以下几种程式：

（1）连续强化：每当出现所期望的行为后都给予强化。这种强化方式最容易塑造良好行为，但一旦取消强化，行为也最容易消退。

（2）定时距强化：是指不管行为本身，每隔一段固定的时间便给予一次强化。例如，员工每个月固定领取薪水和奖金。

（3）定比率强化：是指不管时间长短，每出现固定次数的行为便给予一次强化。例如，业务人员每推销出一定数量的商品便给予一次奖金。

（4）变时距强化：是指强化物按照不固定的时间间隔而出现。例如，不定期检查员工的工作日志，凡是记录详实者便给予一笔奖金。

（5）变比率强化：是指强化物按照不固定的行为出现次数而出现。例如，管理者不定次数地赞赏员工的优异表现。

一般来说，就维持行为而言，变化的、间隔的强化比固定的、连续的强化效果更好，这是因为：一方面它能更经济有效地增加先前行为出现的概率，另一方面它有助于提高强化中断或消失后的抵抗力或坚持性。如果总是连续不断地给予强化，那么一旦强化中断或消失，行为就会立即随之消失；相反，如果强化本来就是变化的、间隔的，那么即便有较长时间的不强化期，行为也能得到维持。根据自己的实验结果，斯金纳指出，总体上看，上述五种强化程式从前到后一种比一种的效果好。

2．强化理论对管理的启示

强化理论给管理工作带来许多重要的启示。一些研究者在斯金纳强化理论的基础上，提出了指导管理实践的行为规则，如哈姆纳（Hamner）曾总结了六项行为规则：

（1）不能以同样的方式奖励所有的人，绩效大小不同，奖励也应不同。如果管理者给每一个人以同样的奖励，其结果是惩罚了工作绩效最好的，奖励了工作绩效最差的。

（2）无反应本身具有强化的效果。管理者可以通过不做某些事情或做某些事情来塑造员工的行为。例如，不纠正一个员工上班迟到的行为，就有可能被认为上班迟到是对的。管理者必须仔细检查他们作出或不作出反应所可能导致的后果。

（3）一定要告诉员工，他们怎么做才能得到奖励。

（4）告诉员工他们正在做的哪些事是错误的。这个信息能帮助下属懂得如何改变工作习惯，否则他们就不会明白为什么奖励被取消或为什么受到惩罚。

（5）惩罚员工时不要当着其同事的面。因为如果这样做了，这个人实际上被惩罚了两次，它会导致下属寻找抵制管理的方式，其结果对在场的人也不好。

（6）使结果和行为相一致。这就要求管理者公正地对待每一位员工。

强化理论是影响和引导员工行为的一种重要方法，通过表扬和奖励可以使动机得到加强，行为得到鼓励；通过批评和惩罚等可以否定某种行为，使不好的行为越来越少。不过，在运用强化理论时，应以正强化为主。同时，无论是表扬还是批评都要在调查研究的基础上，实事求是，及时准确；另外，针对不同的人，强化的方式也应有所不同。

（二）公平理论

员工不是在真空中工作，他们总是在进行比较。如果你刚大学毕业就有人提供给你一份年薪 40000 元的工作，你可能对自己的收入很满意，并且努力工作。可是，假如你工作了一年之后，发现另一位最近毕业的、与你年龄、教育程度相当的同事，年收入为 45000 元时，你有什么反应呢？你可能会很失望。大量的事实表明，员工经常将自己的付出与所得和他人进行比较，而由此产生的不公平感将影响到此人以后付出的努力。

1．公平理论的基本内容

公平理论是美国心理学家亚当斯（J.S.Adams）于 20 世纪 60 年代提出来的，也称社会比较理论。其主要目的是为了解决工资报酬分配的合理性、公平性及其对员工生产积极性的影响问题。

公平理论认为，员工首先考虑自己所得与付出的比率，然后选择与自己年龄、资历等方面相仿的人作为参照对象，将自己的所得/付出比与他人的所得/付出比进行比较。这种社会比较可能产生三种结果，如表 5.4 所示。

表 5.4　社会比较的三种可能结果

觉察到的比率比较			员工的评价
我所得 /我付出	<	他所得/他付出	不公平（报酬过低）
我所得 /我付出	=	他所得/他付出	公平
我所得 /我付出	>	他所得/他付出	不公平（报酬过高）

这里所谓的付出和所得都是个人的主观感觉和判断。付出指的是一个人自己觉得劳动量的多少、效率高低和质量好坏，还包括年龄、性别、所受教育与训练、经验、技能、职

务、社会地位、对组织的忠诚与努力程度等。所得是指一个人得到的劳动回报，包括工资、奖金、地位、权力、待遇、赞赏、表扬、对工作成绩的认可、提升等。当员工感到公平或不公平时，会出现什么后果呢？

一般来说，当员工感到公平时，他们会感到心安理得，保持心理平衡，维持甚或增加工作努力的程度。而当他们感到不公平时，就会产生紧张、焦虑和不安。为了削弱所感受到的不公平以及相应的紧张焦虑情绪，员工就会采取一些行动。具体说来，当员工感到不公平时可能会采取以下几种做法：

（1）采取某种行为改变自己的付出或所得。感到报酬过低的员工可能会减少自己的工作投入，开始迟到、早退、工作不积极等；或者是要求增加自己的所得，如要求增加工资待遇等；如果这些措施都不能奏效，他们也可能做出极端的行为，如挪用或盗窃公司的财物等。而感到报酬过高的员工可能增加自己的投入。

（2）采取某种行为使得他人的付出或所得发生改变。如通过申诉、上告、吵闹等方式，要求降低他人的奖酬，或要求他人增加投入。

（3）在心理上重新估价自己或他人的付出或所得来获得公平感。感到报酬过高的员工更可能采用这种策略，他们常常会通过低估别人的付出或高估自己的付出来获得心理上的平衡。

（4）选择另外一个参照对象进行比较。

（5）辞职。

2．公平理论的实践意义

公平理论在管理实践中具有重要的现实意义。实践证明，该理论所描述的关于公平的感受是一种普遍的心理现象，广泛存在于企业和其他组织环境中，并直接作用于员工的行为过程，影响员工的工作积极性。按照这种理论，管理者应该注意以下几点：

（1）建立奖罚分明的制度

员工的不公平感有时确实是因为组织没有合情合理地奖励员工，存在着“有功者不奖，无功者领赏”的不良现象。组织只有消除这些不合理的现象，建立奖罚分明的制度，贯彻“多劳多得，少劳少得，不劳不得”的原则，增加分配制度的透明度，才能让广大员工真正感到公平。

（2）发展员工参与制度

员工参与有助于保证制度本身的公正性，使员工们感到制度在被严格执行，从而增强公正感，同时还有助于上下级之间的了解，改善上下级关系。企业各项与员工有利害关系的制度（如发展战略、分配制度、奖励制度、晋升制度和考评制度等）的建立，都需要员工参与。

（3）加强组织沟通

沟通是理解的桥梁。上行沟通渠道不畅往往使得员工的不满难以为管理者所了解，下行沟通渠道不畅则使得组织意图不能得到员工的彻底领会。因此，建立畅通的沟通体系，做到信息的上传下达，可以起到疏通“组织经脉”，降低员工不公正感的作用。当然，沟通过程既可以公开透明，也可以照顾到员工的隐私性和安全需要而实行匿名的信息传递。

（4）做好员工的心理疏导工作

公平感具有很强的主观性。人们的公平感受性不同，社会比较的参照标准不同，认识问题的角度和方式不同，对同样的事情也必然会产生不同的公平感。因此，做好员工的心理疏导工作，引导其树立正确的公平观，是非常重要的一项管理工作：首先，要引导员工认识到绝对的公平是没有的；其次，教育员工不要盲目攀比，纯主观的比较往往容易造成不公平感；第三，引导员工成为内控型的人。一项关于员工内外控型与公平感关系的研究发现，外控型的员工更容易抱怨组织不公平，因为他们习惯于把不利的事件归因于外部组织环境；而内控型的员工更容易体验到公平感，他们勇于承担不利事件的责任，积极从自身方面找原因，并努力寻求解决问题的方法。

第二节 中国的激励理论

中国悠久的历史不仅孕育了博大精深的民族文化，而且产生了至今不失光彩的激励思想。中国当代学者也有不少关于激励问题的精彩论述和研究。从我国古代管理思想和当代研究中采撷有关激励问题的花絮，对建立具有中国特色的激励理论有着重要的意义。

一、中国古代文化中的激励思想

（一）“惠民”、“富民”、“爱民”、“教民”的激励思想

儒家主张通过国家的政治、经济、文化、教育等方面的政策来调动人的积极性。孔子提出了惠民、富民的思想，强调为政首先要考虑施恩惠于人民，使人民过上安逸而富裕的生活，否则就是“不仁、不义”。治理国家的目标，首先在于安民。民贫则怨，民富则安。统治者要“使民以时”，不滥征民力，放手让人民从事生产，并做到“薄赋敛”，“节用爱人”，使人民得以“足食”和“济众”，这样会取得“近者悦，远者来”的最佳激励效果。

孟子提出“爱民”、“教民”的激励思想。“爱民”，就是要与民同乐。“乐民之乐者，民亦乐其乐；忧民之忧者，民亦忧其忧。乐以天下，忧以天下，然而不王者，未之有也。”教民就是强调培养人民使命感、责任感、道德感的作用，“善政不如善教之得民也。善政，民畏之；善教，民爱之。善政得民财，善教得民心。”

儒家所阐述的以人为本的仁学思想，应用于现代企业管理中，就是要求管理者要有一颗真诚的仁爱之心。他必须热爱企业，热爱员工，热爱所从事的事业，这种爱心是发自内心的，是真诚的，而不是以小恩小惠笼络人心的权宜之计。要办好企业，就必须得民心，而要得民心，就必须“以仁心买人心”。它的力量是巨大的，对于化解组织矛盾，增进企业凝聚力，激发员工的积极性，具有其他力量不可替代的作用。此外，儒家激励思想用于企业管理工作中，还强调以人为本的伦理道德建设，通过伦理规范和道德教化，培养人们

共同的信念和价值观，增强员工对企业的向心力和凝聚力，激发人的工作积极性，实现企业和谐、有序发展的目标。

（二）因人而异的激励方法

古代的兵书中包含了丰富的激励思想，提倡“以心治心，因人而异，奖罚分明，赏罚及时”的激励方法。

兵家提出，管理者要注重下属的心向、态度，“夫主将之法，务揽英雄之心”。不同的部下，心态、个性各不相同，管理者要因人而异地进行管理、激励。对此，姜尚提出了一些原则：“危者，安之；惧者，欢之；叛者，还之；冤者，原之；诉者，察之；卑者，贵之；强者，抑之；敌者，残之；贪者，杀之；欲者，使之；畏者，隐之；谋者，近之；谗者，覆之；毁者，复之；反者，废之；横者，挫之；满者，损之；归者，招之；服者，活之；降者，脱之。”

《孙子兵法》中指出，奖励时要论功行赏，使奖励具有差别性；奖励要做到公平合理。孙膑认识到有的人贪生，有的人爱财，指出奖励要与受奖人的需要相结合。中国古代兵家还主张根据作战情况的不同，破格奖励，以达到“重赏之下必有勇夫”的效果，“施无法之赏，悬无法之令，犯三军之众，若使一人。”

兵家的激励理论对企业管理中激励方式的选择具有实用价值。企业中员工有个体差异，不同发展阶段或不同类型的企业，员工需求也不同，因此必须因人而异地选择不同的激励方式。随着人们整体物质生活水平的提高，只用物质激励已经不能很好地调动企业员工的积极性，物质激励必须与员工渴望的情感、荣誉、目标、行为等精神激励措施相结合，使员工得到感情上的慰藉、安全感和归属感，才能充分调动员工积极性，实现企业目标。

（三）功利主义的激励措施

管子、韩非子等法家代表人物都认识到人的行为的根本动力是追名逐利，因此直言不讳地提倡功利主义的赏罚理论。管子认为，一般的人都有趋利避害的特性，所有的人，不分贵贱都是“得所欲则乐，逢所恶则忧”，“民予则喜，夺则怒”。追求功利是人的本性，要以利作为杠杆，激励人民的积极性，“得人之道，莫如利之”，“欲来民者，先起其利，虽不召而民自至”。为此，作为统治者必须善于给人以利益，满足人的物质需要。一旦人的利益或需要得到了必要的满足，必将激发起更大的积极性，产生更大效益。但管子不主张无限制地满足个人的私利，他认为应使个人利益的欲望有所节制，否则国家就不好治理，而且对个人私利无所限制，则利益也就失去激励的作用。利益给多了，人们就不当回事了，即“万物轻则士偷幸”，意思是物价低，谋事易，则士人就苟且偷生。

法家激励思想同样适用于现代企业管理。能否正确运用利益原则，处理好管理者和被管理者的关系，在激励中起着重要作用。人的求利思想是行赏有效的思想基础，企业家应该重视和运用这一客观规律，在实践中用好“利”这根魔棒，充分发挥人的潜能与创造力，与员工一起追求企业的最大利益，创造辉煌的业绩。例如，在企业的管理中，将工资、奖金与个人业绩挂钩，就可以激励员工的积极性和责任感。在某些行业，对于脏、累、苦、险的第一线岗位，采取工资福利的倾斜政策，给予各种奖励，也能起到激励的效果。应当充分利用利益的激励作用，激发起管理者和被管理者的智慧与创造力。

（四）赏罚分明的激励约束机制

韩非子把赏、罚作为君主的两根手柄："君以其言授之事，专以其事责其功。功当其事，事当其言，则赏；功不当其事，事不当其言，则罚。"他主张赏罚分明、赏罚公正、赏罚必信、赏罚及时、赏罚有度、随时而变、因俗而动。只要激励得当，用好赏罚二柄，管理工作便容易开展了："政之大纲有二。二者何也？赏罚之谓也。人君明乎赏罚之道，则治不难矣。"

中国古代的管理者善于利用法家的赏罚学说进行名利等方面的赏罚：好名者则赐爵赐号，喜官者则升官，贪利者则赐物；同时，根据被管理者的过失大小，进行不同程度的惩罚。

激励和约束机制是现代管理中引导人们思想及行为的有力武器。古代管理思想中的赏罚分明原则为激励和约束机制的法律化、规范化提供了重要的依据和途径。

二、中国当代激励理论

我国学者在介绍和引进西方激励理论的同时，也在不断构建符合我国企业管理实践的本土化的激励理论。这里我们主要介绍苏东水和俞文钊提出的激励思想。

（一）苏东水的"人为为人"的激励思想

苏东水通过近十年的研究，将中国管理文化的精华概括为"以人为本，以德为先，人为为人"。人为为人，指的是每个人首先要注重自身的行为修养，正人必先正己，然后从为人的角度出发，控制和调整自己的行为，创造一种良好的人际关系和激励环境，使人们能够持久地处于激励状态，主观能动性得到充分发挥。对任何管理者或被管理者而言，都有一个从个人行为逐步向为他人服务转变的过程，即从"人为"向"为人"转变的过程。

"为人"就是管理者要为他人服务，满足员工的生理、心理、物质、情感等各方面的合理的需要，从而从根本上调动员工工作的积极性。"为人"，能使员工感到管理者对自己的重视、信任和接受，从而得到情感上、精神上的激励，并使员工感到自己是组织的一个重要成员，感到自己的参与，感到自己的价值，从而激发积极向上的动机。

（二）俞文钊的激励理论

1．同步激励理论

同步激励理论（synchronization motivation theory）可简称 S 理论。它是指物质与精神激励有机结合、综合使用、同步进行时，才能取得最大的激励效果。用公式表示如下：

激励力量＝Σf（物质激励×精神激励）

这一公式表示只有物质激励与精神激励都处于高值时才有最大的激励力量。两个变量中只要有一个处于低值，都不能获得最佳、最大的激励效果。

该理论否定了单纯使用一种管理方法（用 X 理论或 Y 理论，精神或物质激励措施）的做法，也否定了简单地交替使用 X 或 Y 理论的做法，强调物质与精神激励要同步应用。

2．三因素（激励、保健、去激励）理论

三因素理论又称激励—去激励因素的连续带模式。

三因素具体是指激励因素、保健因素和去激励因素。三者含义上的差别，如表 5.5 所示。

表 5.5　激励、保健、去激励因素的含义

激励因素	保健因素	去激励因素
使人产生满意感	使人产生不满意感	使人产生不满意感
使人提高积极性	保护人的积极性	使人的积极性降低
使工作效率提高	维持原状，不会使工作效率提高	使工作效率降低

激励因素与去激励因素存在于连续体的两个端点，是两种极端的情景。在这两种极端的激励与去激励因素之间，还应该存在许多种强弱不等的激励形式，它们构成一个连续带。保健因素位于激励—去激励连续模式的中间过渡地带。这些因素之间是可以相互转化的。

在企业管理中，管理者要正确区分哪些是激励因素，哪些是去激励因素，然后创造条件，使去激励因素向激励因素转化。

3．公平差别阈理论

俞文钊等人在亚当斯公平理论的基础上提出了公平差别阈这一理论模式。亚当斯认为，人们总是将自己所做的贡献和所得的报酬与别人进行比较。如果两者之间的比值相等，双方就会有公平感。但在现实生活中，既存在着两个人的条件相等的情况，也存在着条件不相等的情况，如资历、工龄、职务、劳动投入量等方面的差异。在这种情况下，无差异分配不仅不能产生公平感，反而会产生不公平感。在两个人之间的条件不相等时，适宜的差距分配才能使人产生公平感。

公平差别阈是指能使两个条件不相等的人，刚能产生公平感时的适宜差别的比值。这是一个可以测量的值，这一概念与量值适用于分配领域的各个方面，如工资、奖金及其他各种形式的分配。

俞文钊等人的实验结果表明，承包者与员工之间的报酬的公平差别阈为 1∶2～1∶3，即承包者与员工的收入差别最多为 2～3 倍。如果大于此值，员工是不能接受的；如果小于此值，承包者也是不能接受的。

根据这一理论，管理者在进行分配时，既要打破平均主义大锅饭的分配方式，也要防止差距悬殊的分配方式，坚持适宜的差距分配，提高员工的工作积极性。

第三节　激励理论的比较及其文化相对性

一、激励理论的比较

前文介绍的各种激励理论的侧重点各不相同，下面对西方的激励理论做一简单的比较，如表 5.6 所示。

表 5.6 西方激励理论的比较

西方激励理论	类 型	主 要 观 点
需要层次理论	内容型	当员工低层次需要满足后，就努力追求高层次需要的满足
双因素理论	内容型	激励因素（工作内容或工作成果）是导致满意的因素，保健因素（工作环境等）是导致人们不满意的因素
目标设置理论	过程型	清晰、恰当的目标具有激励作用，能够提高员工的工作绩效
期望理论	过程型	只有当人们认为实现预定目标的可能性很大，并且又具有很重要的价值时，该目标对人的激励作用才最大
强化理论	过程型	人的行为由外部因素控制，行为是其结果的函数
公平理论	过程型	当员工自己的收入/付出比与他人的收入/付出比相等时就感到公平；不公平出现后，员工会采取行动消除这种不公平

从表 5.6 可以看出，每种理论关注的重点是不一样的。需要层次理论和双因素理论属于内容型激励理论，因为它们都着重于探讨激发和引起人们行为动机的各种因素，而不关心动机产生的过程。其中需要层次理论中的生理需要、安全需要和社交需要相当于双因素理论中的保健因素，而尊重需要和自我实现需要相当于双因素理论中的激励因素。目标设置理论、期望理论、强化理论和公平理论都属于过程型激励理论，因为它们都试图揭示行为动机产生的过程，着眼于分析和探讨各种因素是如何交互作用，从而引起特定行为的。其中，目标设置理论、期望理论和公平理论都属于认知的过程激励理论，它们的共同之处是：认可个体和环境之间的差异，注重个体认知在行为产生过程中的作用，承认行为的可预测性；不同之处在于，它们各自所强调的认知过程有别。目标设置理论强调目标和任务本身的激励价值和人们对其可接受程度的认知，而不关注行为后果的价值；期望理论强调人们对目标实现可能性的认知评估及对目标实现后所带来价值的认识，公平理论强调社会比较过程及其带来的心理和行为后果，这两种理论都强调行为后果的激励价值。强化理论属于行为的过程激励理论，强调先前的行为后果本身（而非个体对行为后果的预期和评价）对后继行为的强化作用，认为行为是由其结果塑造的，否定了个体认知过程的作用。

每种激励理论都有其适合说明和解决的问题，也都有一定的局限性。有人说："激励理论是个筐，什么东西都可以装。"每个组织都应该根据自身的发展状况和存在的问题，灵活地选用适合自己的激励理论，来指导自己的管理工作。但对于调动员工的工作积极性来说，以下几点具有普遍的意义：重视激励的积极作用；承认员工不同的能力、期望和效价；建立明确的绩效—回报联系；提高环境的整体质量；尽可能让员工参与决策。

二、激励理论的文化相对性

民族文化的差异性是应用激励理论时需考虑的重要问题。大多数的激励理论都是由美国心理学家建立，并在美国文化背景下得到验证的。追求自我利益是这些理论的出发点和前提，资本主义和个人主义是这些理论的文化导向，因而它们可能更适用于诸如英国、澳

大利亚等具有相同文化背景的国家。而对于集体主义占优势的国家（中国、新加坡、日本和墨西哥等），更看重的是个人对组织或社会的忠诚，组织的进步和发展是他们关注的重心，因此应用这些理论时，需要进行适当的调整，如强调以集体为基础的职务设计、群体目标和群体绩效评估等。如果过于强调对个人表现的奖励，则会产生适得其反的效果。

有一个例子很好地说明了激励理论的文化相对性。在日本工作的一位加拿大籍经理决定将一位表现优异的年轻日籍女销售代理提升为经理。但出乎意料的是，升职非但没有激发出这位新经理更好的工作表现，还让其原有的工作热情也丧失了。为什么会这样呢？因为日本人对于与同事之间的和谐关系和协调配合有很高的需求，升职作为个人奖励使得升职者与其同伴之间产生了隔阂，并始终困扰着她，从而压抑了她的工作热情。

我国受儒家文化的影响很深，重视血缘关系、强调家族主义传统、崇尚集体主义精神是我国传统文化的显著特点。因此，在使用西方的激励理论时，必须考虑到这些状况，进行适当调整。在借鉴西方激励理论基础上，根据我国的国情和实际，探索具有中国特色的激励理论是未来研究的方向。

本章小结：

1．西方的激励理论主要包括六种：需要层次理论、双因素理论、目标设置理论、期望理论、强化理论和公平理论。它们从不同角度分别回答了以下三个问题：（1）是什么激发或驱动行为的？（2）是什么引导行为方向的？（3）是什么使行为得到维持的？

2．马斯洛的需要层次理论和赫茨伯格的双因素理论是内容型激励理论的主要代表。前者试图从生理需要、安全需要、社交需要、尊重需要与自我实现需要及其相互关系的分析中找到工作积极性的基础，后者则从激励因素和保健因素两个方面来分析员工工作积极性的源泉。

3．目标设置理论、期望理论、公平理论属于认知的过程激励理论。目标设置理论通过提供具体的、富有挑战性的目标激励员工，引导和促进他们的工作积极性。期望理论强调只有当人们预期到某一行为能给个人带来既定结果，且这种结果对其具有吸引力时，才会积极地从事这种行为。认知的作用主要在于把握以下三种联系：努力与工作绩效的联系、工作绩效与奖赏的联系、奖赏与满足个人需要的联系。公平理论着眼于个体在与类似的人进行比较时所产生的公平认知及其产生的心理和行为后果。而强化理论属于行为的过程激励理论，强调环境因素和行为结果本身（强化）对后继行为的激励作用，否定个体认知评价的作用。

4．中国古代文化中蕴涵着丰富的激励思想。儒家提出“惠民”、“富民”、“爱民”、“教民”的激励思想；兵家提出了因人而异的激励方法；法家提出了功利主义的激励措施和赏罚分明的激励约束机制。这些激励思想对于指导现阶段我国的组织管理实践依然具有重要的价值。

5．我国当代学者在介绍和引进西方激励理论的同时，也在不断构建具有中国特色的激励理论。苏东水提出的“人为为人”的激励思想，俞文钊提出的同步激励理论、三因素理

论和公平差别阈理论是其中的典型代表。

6．任何理论的发展都离不开特定的文化背景。在应用西方的激励理论时，应考虑我国的国情和文化特点，同时也要结合我国的实际不断探索适合我国组织特点的激励理论。

本章思考题：

1．有人说现在的员工比以前任何时候都更难激励。你同意这一看法吗？为什么？

2．想一想过去你受到最大鼓舞和激励的一项工作或任务和受到最大挫折的一项工作或任务，用本章介绍的激励理论来解释其中的原因。

3．假如你是一个团队的领导者，你发现团队中有位员工对团队项目不愿付出更多的努力，因为她认为自己没有能力为团队做出贡献。请你根据期望理论，提出激发这位员工工作积极性的措施。

4．在一些非洲国家，许多员工都认为薪水的多少，一定程度上是由员工拥有的孩子数量决定的，因为孩子多的员工需要更多的钱来抚养家庭。请你根据公平理论谈谈对这一现象的看法。

案例分析 1：

沃尔玛成长壮大的魔幻组合

当萨姆·沃尔顿（Sam Walton）于 1992 年去世时，一些业内人士怀疑他在 30 多年前创建的沃尔玛连锁店作为一种打折零售商能否保持原有的业绩。在 20 世纪 90 年代中期，沃尔玛的股票像其他公司那样失去了牛市的增长状况，那些业内人士的怀疑似乎是正确的。然而今天沃尔玛又重新振作起来了，不仅在 8 个国家及美国 50 个州有了分店，而且在同类折扣商店中，销售业绩处于领先地位。

许多人相信沃尔玛成功的关键之一是它懂得如何去激励其销售队伍。例如，沃尔玛商店的每一次员工聚会都是令人振奋或鼓舞人心的——这种聚会是多年以前沃尔顿亲自发起的代表性事件或典型事件之一。当销售人员为一天的购物狂潮做准备时，商店经理带领员工高呼“给我一个 W，给我一个 A，给我一个 L……”，以此来激励销售人员进入工作状态。此外那些职员还了解他们的顾客想要什么以及有多少人正在购买他们的商品。然而，为了使他们更加明确，职员们仍然要知道精确的销售数额以确切表明他们某一连锁店的运营状况。例如，与前一天、前一个星期或前一年相比较他们赚了多少钱？哪些是热销产品以及它们的利润是多少？各部门的代表都可以骄傲地宣布问题的答案，但不是使用大多数商务会议上那种枯燥的语调。在沃尔玛，会使每个人都对回答此类问题感兴趣。例如，你是一名沃尔玛的员工，你可能由于不久前猜中了前 4 个星期商店售出 15850 包商品而赢得一包奥利奥饼干。的确，这样的奖品对于员工来说是微不足道的，但它能增进沃尔玛员工之间的友谊并使得在此工作的经历更有趣味性。

问题：

沃尔玛的成功使它赢得了世界最大零售商的地位。除了以诱人的价格、销售质优的产品这一关键因素之外，沃尔玛还有什么因素值得我们借鉴？

案例分析 2：

小王的变化

小王在一家公司的会计部工作。刚开始工作的时候，她工作热情非常高涨，尤其令她感到高兴的是，她发现了改善本部门工作流程的办法，并且向上司提出了许多建设性的意见。上司也很赞同这些方法，甚至在部门内贯彻了一些方法。但是，这位上司似乎没有时间对小王表示感谢或者是解释为什么有些建议没有被采纳。事实上，这位上司对本部门的任何做法从来不会向任何员工解释什么。

第一年底，小王根据上司对她的绩效评估，获得了 6% 的加薪。本部门的其他 11 位员工也获得了同样的加薪。小王感到很沮丧，她不知道下一年她要怎样做才能获得更高的加薪。同时令她沮丧的是，一位新进来的员工小张居然得到了本部门最高的加薪（10%）。小张工作能力平平，和他一起工作的同事也不喜欢他，根本没有理由得到这么高的薪水。然而，小张显然已经成为上司最喜欢的员工了。

第二年，小王对这家公司的热情下降了很多。她不再是第一个到达办公室的人，也不再为公司提出任何建设性的意见了。她甚至请了好几天的病假去拜访朋友，而在第一年她只请了 2 天假，并且好几次医生要求她休息，她都没有休息而是继续工作。又到年底了，小王不知道自己的薪水会涨多少，但是她已经开始考虑辞职了。

问题：

1．在本案例中存在什么问题？

2．你认为产生这些问题的原因是什么？

3．组织应该采取什么措施纠正这些问题？

技巧库：

目标设置真的有效吗？

具体、有难度的目标能提高工作效率，下面设计的练习会让你亲自体验一下。你所需要的是愿意合作的一个班级的学生以及一些简单的设备。

1．指导语

（1）从一本书中随机挑选一页并打印几份，仔细计算一页上的单字的数量。

（2）找一个有 30 个或更多学生的班级，他们对目标设置一无所知。根据随机原则，将这些学生平均分成三组。

（3）要求第一组——“基线”组的学生尽可能将这页纸上的内容抄到另一张纸上，并给定他们时间，指导他们动作要快一些。确认学生抄写单字的最高数量，然后将这个数字

乘以 2，作为具体的有难度的目标水平。

（4）要求第二组——“具体目标”组的学生在同样的时间抄写这些单字，让他们达到步骤（3）中的具体目标。

（5）对于第三组——“竭尽全力”组的学生，要求他们重复同样的过程，但不给他们规定具体的目标，而是指导他们“尽最大努力去完成任务”。

（6）计算“具体目标”组和“竭尽全力”组学生抄写单字的平均数目，进行适当的显著性检验（本例中用 t 检验）以决定不同成绩水平差异的显著性。

2．问题讨论

（1）两组学生成绩是否存在统计上的显著性差异？假如存在的话，“具体目标”组的学生是否如期望的那样优于“竭尽全力”组的学生？就目标设置的效果性来说，这种现象揭示了什么？

（2）假如预测的结果得不到支持，你认为为什么会发生这种现象？在这个过程中，是什么导致了这样的失败？是具体目标（例如“基线”组最快速度的两倍）太高以至于遥不可及？或者是目标定得太低，实现它太容易？

（3）假如目标降低或提高，会发生什么情况？

（4）你认为提供关于目标达成情况（如有专门的人数出抄写单字的数目，同时报告给正在抄写的人）的反馈有效吗？

（5）你相信目标设置对于其他类型的任务同样有效吗？尤其你相信目标设置能提高自己在某些方面的绩效吗？请解释这种可能性。

第六章　激励理论的应用

在第五章中，我们对西方和我国的激励理论有了较为系统的了解。实际上，激励是一项系统工程，要想起到激发动机的作用，必须从满足员工的需要出发，帮助员工建立合理期望，设置合理目标，提高员工的公平体验，从多角度、全方位进行规划和设计。那么，具体应该怎样实施呢？在本章中，将从薪酬体系设计和工作设计两方面，对激励理论在实践中的应用问题进行探讨，并对激励实施之后的结果进行分析。需要说明的是，激励产生的心理和行为后果很多（如组织公民行为、工作投入、工作满意度、离职率、组织承诺等），本章仅选取组织公民行为和工作投入这两种最直接的后果进行分析，其他后果将在第十二章进行深入分析。

第一节　激励的途径和方法

激励是一个体系或系统，它贯穿于管理的全过程，依托各种管理制度、管理工具来实现。激励是稳定存在的，它是每个企业成长发展必不可少的一部分；激励也是动态变化的，不同企业在发展的不同阶段，针对不同员工所采用的激励措施应该是不同的。因此，真正有效的激励是在综合运用各种激励理论的基础上，对各种激励措施进行选择、组合及优化的过程。本章将从薪酬激励和工作设计两方面，对企业中实际应用的激励途径和方法作一介绍。

一、薪酬激励

（一）薪酬的含义及类型

薪酬是员工因向所在组织提供劳动或完成任务而获得的各种形式的酬劳或酬谢。它是劳动力价格的支付形式，在市场经济环境下，同时又是人力资本竞争的价格表现。按照存在的形态不同，薪酬可分为经济性薪酬和非经济性薪酬。经济性薪酬又叫外在薪酬，是指以物质形态存在的各种薪酬。它又可分为直接薪酬和间接薪酬两大部分：直接薪酬包括员工收到的基本工资、奖金、津贴、加班费、佣金、股权等；间接薪酬也叫福利，包括所有未包括在直接薪酬中的货币薪酬。典型的福利项目包括：带薪休假、各种保险、子女入托入学、父母托管等。非经济性薪酬是指员工由工作本身带来的满足感，包括参与决策、较大的工作自主权、工作认可、挑战性工作、良好的工作环境、个人发展机会、职业保障、荣誉、自尊等。具体关系如图 6.1 所示。

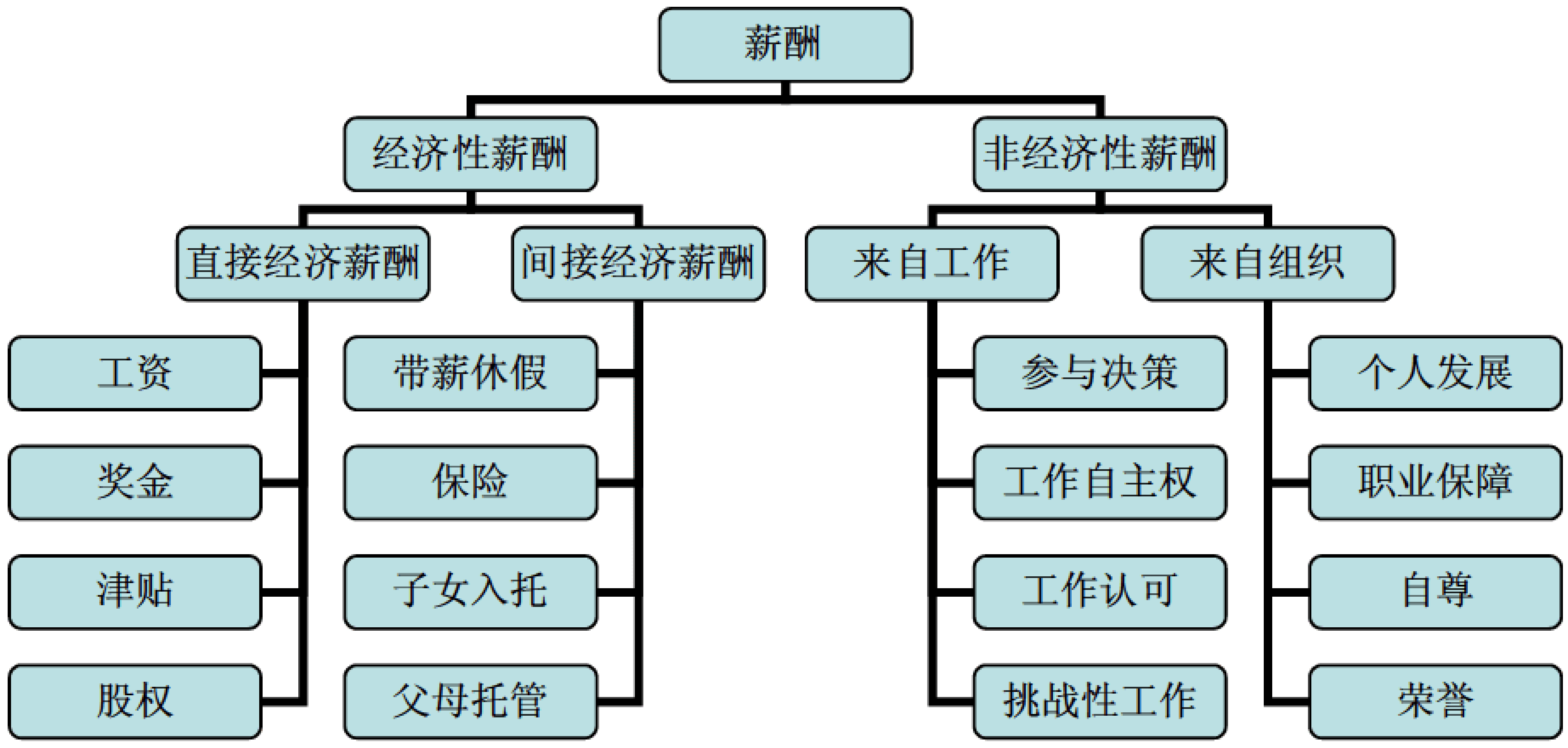


图 6.1　薪酬结构图

（二）薪酬激励的作用及原则

薪酬是激励员工的最重要途径之一。对员工而言，薪酬的功能不仅在于能满足个体及其家人最基本的物质需要，还在于它的高低是员工自身价值的体现；对企业而言，薪酬是最重要的人力资源管理手段之一，将经济性薪酬和非经济性薪酬有机结合，可以影响员工工作绩效和满意度，起到良好的激励效果。

在运用薪酬对员工实施激励时，应遵循以下几个原则。

首先，薪酬激励的公平性原则。中国有句古话：“不患寡而患不均”。公平是薪酬激励的核心问题。企业在进行薪酬设计时必须考虑如何建立一个适应本企业特点的薪酬体系，尤其要注意在企业目标指导下的薪酬公平性。

公平性分为外部公平性和内部公平性。外部公平性是本企业员工把自己的工资水平同其他同类企业的工资水平相对比而得到的结果。对该结果的判断可以作为求职者是否接受该工作的重要参考指标，也可以对在职者的工作积极性和满意度产生正面或负面的作用，从而影响其做出去留决策。市场薪酬调查是实现外部公平性的前提和重要手段。内部公平性是企业员工同本企业内部的其他员工进行比较的结果，它分为横向公平和纵向公平。横向公平比较的对象是从事与自己相同工作的同事，或者其他工作岗位上与自己级别相同的员工。纵向公平比较的对象则是企业内部与自己级别不同的员工。比较的结果将会影响到员工整体的工作态度、献身精神、晋升吸引力以及调换到其他岗位上意愿的强烈程度。做好岗位评价、制定合理的工资结构和等级级差是实现内部公平性的关键所在。

其次，薪酬激励的个性化原则。任何一套激励方案，除了整体上要系统设计，对组织成员的激励具有统一性外，还要遵循个性化原则，根据不同员工的不同需求状况，提供个性化的人力资源解决方案。一些管理者错误地认为钱可以激励所有的员工，把涨工资、发奖金当作放之四海而皆准的“法宝”。实际上，在不了解员工需要的情况下一味提升工资水平只会把整个公司的工资体系搞乱，引起其他员工的不公平感。还有些管理者采用“大一

统”的激励策略，简单地把20岁和40岁员工的激励措施混为一谈。实际上，职业生命周期的变化会带来个人追求的变化，针对不同年龄阶段的人，应该采取不同的激励方式。20岁的员工头脑灵活，精力旺盛，赋予他们富有挑战性的工作可以最大限度地激发他们的热情。而40岁的员工需要的是工作的稳定，为他们解除后顾之忧，帮助他们达到家庭与事业的平衡可能发挥更有效的激励作用。

最后，激励的优化组合原则。一个好的激励方案，应该在目标的指引下，做到物质激励和精神激励相结合、外部激励与内部激励相结合、正面激励与负面激励相结合。采用哪些激励方法的组合要视员工实际情况而定。只要最终能够同时起到“保健”和“激励”的双重作用，就不失为一个成功的激励方案。

（三）薪酬体系设计

薪酬激励的方式经历了从传统的薪酬体系设计，到目前宽带薪酬、股份奖励等新形式的演变。传统的薪酬体系主要由固定部分的基础工资和浮动部分的业绩工资组成。基础工资以职位、个人特征（如学历、技能等）为支付依据，它的满足可以使员工的职业安全感得到保障，可以更有效地转化为工作投入度。业绩工资则主要以绩效为支付依据，具体形式包括奖金、津贴、股权奖励等。

1．传统薪酬设计的一般步骤

（1）岗位分析。把每个岗位的工作职责、任务目标、汇报关系、任职资格等要素理清楚，形成规范的岗位说明书。在此基础上，形成岗位评估，理清岗位之间的相对价值，拉开不同岗位不同职级之间的距离。

（2）薪酬管理。进行广泛的薪酬调查，对所在地区同行业、同级别的相同岗位进行薪酬比较。同时考虑本企业有哪些优势岗位和核心员工，有哪些一般岗位和普通员工，给予前者高于市场水平的薪酬以吸引和留住人才，给予后者与市场价格相当的薪酬以控制人工成本。这样准确地量化各岗位的价值，确定其具体的固定工资水平，形成一个科学的、“对内公平，对外有竞争力”的固定薪酬体系。

（3）绩效管理。绩效管理第一位的工作就是制定好各部门、各岗位的绩效指标体系，并将他们落实下去。其次是切实公平地将绩效结果体现到反馈中去，包括员工对个人绩效水平的知情权，以及在业绩工资中体现绩效的差异。这样形成科学的岗位浮动薪酬体系，才能真正激励员工。业绩工资占整体薪酬的比例不能太小，否则起不到激励作用，企业内部仍然会回到平均主义“大锅饭”的老路上去；比例也不能太大，否则会冲淡了基础工资的“保健”作用，业绩工资的“激励”作用也不能很好地发挥。

需要强调的是，上述三个步骤是一个整体，没有科学的岗位分析，薪酬管理和绩效管理就失去了安身立命的根本；没有绩效管理，仅靠岗位分析和薪酬管理，激励也就成了“水中月，镜中花”。因此，三者是缺一不可的，必须相互配合才能真正起作用。

2．一些新型的薪酬形式

除了传统的薪酬设计以外，薪酬激励还出现了一些新的形式。这些形式作为传统方式的有益补充，已经在许多企业中实行开来，起到了极其重要的作用。

（1）宽带薪酬体系。在传统的薪酬体系中，往往出现这样的问题：一个平时表现突出、

业绩优良的员工，即使苦干一辈子，如果没有空缺的提升职位给他，他也永远没有一个业绩平平的上级主管工资高；一个在本职工作岗位上表现出色的员工，如果其能力水平不适合提升到更高的职位上，其薪酬就不能得到大幅的增加。因为传统的薪酬体系总是与职位层级紧密挂钩的，而且层级数目可能很多，职位层级相差大，其薪酬等级也相差大。

宽带薪酬可以有效地解决上述问题。所谓宽带薪酬是对传统的带有大量等级层次的垂直型薪酬结构的一种改进或替代。根据美国薪酬管理学会的定义，宽带薪酬是指对多个薪酬等级以及薪酬变动范围进行重新组合，从而变成只有相当少数的薪酬等级以及相应的较宽薪酬变动范围。也就是说，宽带薪酬拥有较少的薪酬级别，较大的级别内部差别。

宽带这个概念来源于广播术语。宽带中的“带”是指工资级别，“宽”是指工资浮动的范围比较大。在这种薪酬体系中，员工不是简单地按照企业的薪酬等级层次垂直往上走，薪酬也不是简单地随着岗位的变化而变化，而是身处同一个薪酬宽带中，职级不变，只要自身能力、绩效改变，薪酬就会有很大不同。有时同一宽带的最高值与最低值之间的区间变动比率可达到 100%或 100% 以上，所以，在这种薪酬体系下，相同岗位上的员工薪酬可能迥然不同；较低职级的员工获得与上级主管相当甚至更高的薪酬也不足为奇。宽带薪酬更注重的是绩效，从传统的注重岗位薪酬转变为注重绩效薪酬，带宽拉大，员工的薪酬更加灵活，激励效果也更加明显。

宽带薪酬管理模式的有效实施，有赖于几方面的条件：首先，宽带薪酬支持扁平型组织结构。宽带薪酬并不适合所有的组织结构，它是为配合扁平型组织结构而量身定做的。只有战略明确、组织层级少、组织结构与发展战略相匹配、治理完善的企业，才适合采用宽带薪酬。那些薪酬管理和人力资源体系的基础非常薄弱，没有规范的职位说明书以及相应的岗位评价的企业，即使实施宽带薪酬，也只会造成一片混乱的局面。

其次，宽带薪酬要求高素质的薪酬管理人员。实施宽带薪酬提高了对人力资源管理部门的要求，同时也加强了各部门之间的密切合作。由于宽带薪酬高度依赖于绩效评价，因此，就要求企业做好岗位分析、绩效考核等方面的基本工作，要求薪酬管理人员以专业的水准，公平的态度为企业员工服务。这对企业人力资源管理部门，尤其是薪酬管理人员提出了较高的要求。

第三，宽带薪酬注重沟通。宽带薪酬的实施可能使领导者的权威因为薪酬不及下属而受到挑战。这就要求企业的管理层和员工能及时全面地进行沟通，使宽带薪酬的内涵得到组织成员广泛而深刻的理解，以一种正确、理性的态度对待薪酬差异。

（2）员工持股计划。企业是由经营者和员工共同组成的。企业效益的好坏不仅有赖于经营者大政方针的正确性，更与员工的努力程度是分不开的。因此，企业激励的对象不能只局限在经营者、管理层和个别“精英”，还应惠及广大员工群体。员工持股计划就是专门针对企业员工的激励而设计的。

员工持股计划（Employee Stock Ownership Plan，简称 ESOP）是指企业内部员工以个人出资（大都指未来劳动的收益）的方式或贷款融资的方式，来持有本企业的一部分特殊股票，并委托员工持股会集中管理的一种产权组织形式。员工持股计划本质上属于一种股票奖励计划，员工可借此渠道参与企业经营管理和分享利润。

员工持股计划的根本出发点基于这样一条基本原则：人们对于自己的东西总是会更在乎些。当员工仅仅是为薪资工作的时候，他们没有办法，也没有热情发挥自己的创造力、持久力和主观能动性。而一旦员工拥有公司的股份时，作为股东，他们就会将企业利益与个人利益紧密相连，会直接关心生产效率和产品质量的提高。有人曾用“四金”来形容员工持股计划：金手铐、金色的梦想、金色的握手和金饭碗。“金手铐”是指员工持股计划可以用企业发展所带来的丰厚收益把人“铐住”，使员工为企业的发展贡献力量；“金色的梦想”是指股份或者期权制是一种长期的报酬，具有诱人的长远利益；“金色的握手”是指员工持有企业股份后，企业经营者、管理人员和普通员工都成了“同一条战壕里的战友”；“金饭碗”是指实施员工持股计划后，员工会加倍珍惜自己的工作机会，离职率会相对降低。

员工持股计划对于不同的公司类型具有普适性，无论知识密集型的高科技企业，还是劳动密集型的零售业、制造业，无论上市公司（公众公司），还是非上市公司（私有公司）都可以使用；员工持股计划对于不同员工具有普惠性，它面向本企业的全体员工，尽量扩大持股面。但是普惠性并不是说在个体间平均分配股份，不同员工间持股的数量差距一定会不同。

员工持股计划在推行过程中需注意一些问题：首先，该计划并不能解决企业面临的全部问题。对那些产品无销路或者销路不畅的企业，特别是属于夕阳产业的企业，如果不能在产品和经营方向上走出新路，仅靠实行员工持股计划带来的资金成本的降低和员工对企业关心程度的提高，只能延缓企业存活的时间，而无法从根本上改变破产的命运。其次，该计划的福利化和短期行为化将会降低激励效果。一些企业在公司上市之初，将一部分股票送给员工持有。而员工往往利用内部员工股和市场股的价差，很快将股份抛售变现。这无形中助长了上市公司视员工股为一种短期福利的倾向，违背了该计划加强企业内聚力、调动生产积极性的初衷。同时，大量员工股被非本企业员工持有，造成员工股份的外部化，容易造成企业被外控股或收购。

二、工作设计

（一）工作设计的含义及其发展过程

所谓工作设计是指对工作完成的方式以及某种特定工作所要求完成的任务进行界定的过程。通过工作设计可以知道提高组织绩效的最佳办法，并同时可以满足员工成长与福利方面的要求。

工作设计的发展过程分为两个阶段。第一个阶段是以泰勒的科学管理为核心的专业化的工作设计，它以任务导向为特征，主要关心工作效率，忽略了“人”的因素在工作中的作用，这种工作设计不具有激励的价值。最具有代表性的是“泰勒制”。它对每个工种进行了精确的时间—动作分析，确定一名工人在最适合的条件下所能完成的最大日工作量，并以此为基础制定工作定额，对工作进行科学的设计。这种工作设计的结果是工作效率有了明显的提高，但员工成了名副其实的“工作机器”，员工的不满大幅增加，劳资关系急剧恶

化。直到 1911 年，工厂工人的大罢工警醒了管理专家们，开始重新审视工作问题。

第二个阶段的工作设计又称工作再设计，是为了适应发展的需要，对某种工作的任务或者完成任务的方式做出改变的过程。现代的工作再设计反对“泰勒制”对人性的泯灭，以 “人”为导向，强调工作中人际关系、组织气氛、员工士气以及管理方式的重要作用，目的在于提高组织绩效的同时，提升员工满意度，具有重要的激励价值，是激励员工的重要途径之一。

（二）工作再设计的几种常用形式

1．岗位轮换

所谓岗位轮换，就是将员工由一个岗位调到另一个岗位以扩大其经验的方法。传统的工作设计强调劳动力的“专业化”，它用严格的标准、科学的方法将员工训练成为本行业的“能手”后，就将该员工与岗位的匹配固定下来，希望他熟能生巧，创造更高的效率。然而事与愿违，这些“能手”的工作效率在达到一定水平后，往往产生“高原现象”，驻足不前，有的甚至会产生一定程度的倒退。调查发现，一味的“专业化”所带来的“单一化”是员工厌倦感的来源，也是企业整体效率下降以及企业内部缺乏活力的根源所在。而岗位轮换恰好能够解决这个问题。

IBM 公司大中华地区总裁周伟琨连续工作三十多年不易其主，有人问及原因，他这样答道：“IBM 给你提供了很多机会去尝试新的工作，我在 IBM 的头 25 年里，平均每两年换一种工作，而且工作地点遍及中国大陆、香港、台湾、日本、澳洲，每种新工作都是一个新挑战，试想，还有哪家公司能为你提供这么多机会，这么大的空间呢？”周伟琨的例子生动地向我们展示了岗位轮换所带来的积极作用。它一方面可以减少员工由于长期从事一项工作而产生的单调感、厌倦感，让他们体会到工作带来的新鲜趣味，保持工作活力；另一方面岗位的变换不仅可以帮助普通员工更深刻地了解自己，找到更能发挥优势的工作岗位，还能使能力突出者全面了解公司的运作流程，为提拔进入高级职位做好经验准备。当然，岗位轮换也会带来一些弊端。把员工从一个岗位转入一个新的岗位，培训成本必然相应增加。而且轮换过频，员工从事每种类型工作的时间过短，工作无法深入，必然导致整体工作效率和绩效的下降。

2．岗位扩展

岗位扩展又称工作扩大化，是指通过增加岗位的工作内容，使一个员工同时承担几项工作，改变原来工作范围窄、简单重复的情况，形成广泛的工作范围和较少的工作循环重复的一种工作设计方法。岗位扩展是工作广度的横向扩展。原本一名工人仅负责拧螺丝，而岗位扩展则会把整个上底盘的工作都交给他负责，那么一项工作变成了数项工作。工作种类多了，所需的动作技能多样化了，工作更有意义了。

岗位扩展并不是随意的扩展，给员工增加的工作一般都与其先前工作在性质上相似、程序上相连，这样可以使员工不经培训即可胜任，节省了招聘新员工所需的培训费用；工作程序上的连贯性则避免了产品或任务在不同员工之间的传递，减少了交接的程序，节省了时间。另外，从员工成就感角度而言，以前所从事的仅仅是一大项任务中很微小的一部分，个人在整个工作中的地位感不清，而进行了岗位扩展之后，员工完成的是一个较大的

单元，甚至整个产品，这对员工的个人价值是一个很好的承认，可以极大地激发员工掌握更多知识和技能，丰富工作经验。

当然，岗位扩展也存在一些弊端。对于那些需要层次较低的员工，获得更高的报酬是他们工作的唯一目的，他们并不认为增加一些额外的工作是权力和责任的象征，而是把它当作一种额外的负担，不仅起不到激励效果，还会助长负性情绪的产生。岗位扩展使员工疲于更换工作项目，单一的动作变得复杂也是许多员工不愿看到的。因此，在实施岗位扩展之前，调查员工的实际需求，因人而异、对症下药是十分重要的。

3．工作丰富化

工作丰富化是指通过工作内容和责任层次的基本改变，使员工在计划、组织、指挥、协调、控制等方面承担更多责任的工作设计形式。工作丰富化是工作的纵向扩展，它不仅给员工分派了更多的工作任务，还为员工提供了获得更多赏识、进步、成长和职责的机会。在挑战性工作和自主性工作氛围的双重刺激下，员工可以更好地发挥主观能动性，更优异地完成任务。

授权和信任是工作丰富化有效实施的前提。工作丰富化要求每一名员工都必须自行规划、设计自己的工作，自行控制生产的速度和品质，自行负责工作的成果并承担相应的责任。在这种高度“自治”的背后，管理者的授权和信任是必不可少的。在承担责任的同时权力充分下放，相信员工的个人能力和优秀品质可以圆满完成任务，不时时监管，不横加指导，让员工成为工作的主人。

工作丰富化也有一定局限。更多的工作职责就意味着更多的知识和技能，因而企业必须增加培训成本以及整修和扩充设备的费用，还需要付给员工更高的报酬。这就要求企业有相应的薪酬体系和良好的工作环境。没有这些辅助设置，而一味增加员工的工作内容和层次，只会起到相反的效果。

4．弹性工作制

弹性工作制是指在完成规定的工作任务或固定的工作时间长度的前提下，员工可以灵活自主地选择工作的具体时间安排，以代替统一、固定的上下班时间。弹性工作制是对传统工作安排进行的重组或再设计，员工可以用这种新型的日程安排，在不损失工作时间的情况下，满足个人多样化的时间需要，包括履行家庭职责、日常生病求医、社交活动等。

弹性工作制的实施有如下几种形式：

（1）核心时间与弹性时间结合制。这种形式的弹性工作制主要由核心时间、带宽时间和弹性时间组成。核心时间是每天的工时中所有员工必须到班的时间，这个时间段里可能会有会议安排，或是重大事件需要集中处理。带宽时间界定了员工最早到达和最晚离开的时间，核心时间被包括其中。弹性时间则是员工根据个人需要，可以自由选择的时间，只要全部工时得到完成，每天的弹性时间可以有不同。这种工作制主要被小型公司采用。

（2）成果中心制。这种形式的弹性工作制是以任务的完成为指标的，员工只需在所要求的期限内按质按量完成任务即可获得薪酬，具体的时间进度安排可根据个体差异，将工作活动调整到身心状态最佳、最具生产效率的时段内进行。

（3）紧缩工作时间制。这种形式的弹性工作制可根据员工个人实际能力，通过增加每天的工作时间长度，使一个完整的工作周在少于五天的时间内完成。剩余时间或休假或娱

乐，由自己处理。

（4）全日制工作与临时雇员队伍相结合制。目前一些企业正在向“双轨雇佣制”的方向发展。其中，核心轨道是全日制的正式雇员队伍，辅助轨道则是机动灵活的临时工队伍，两者互相配合。

弹性工作制的优点可以从三个角度进行理解：对于员工而言，灵活的时间使员工对个人的工作安排有了更大的自主权，在处理工作—生活平衡时有了更大的机动性，自尊、社交需要得到满足，满意度更高；对于雇主而言，不必苦心积虑于员工的缺勤和迟到现象，公司设备更优化地使用，办公资源紧张将得到缓和，利用了员工的高峰时间优势，避免“出工不出活”的尴尬局面；对于顾客而言，企业拉长了工作时间，更多的人可以接受到服务，企业形象得到提升。

弹性工作制也存在一定局限：首先，在该制度建立之初，必须对该岗位的工艺流程和技术规范进行严密的考察和规划，能进行精确的个体工作绩效（质量、数量）考核的工作才适宜实行该制度；其次，必须考虑到监管上的安排，要确保有充足的员工可以轮班，有良好的沟通协作渠道避免“盲点”，有严密的管理规章制度保证实施的井井有条；第三，一些岗位的特殊性使它无法实施弹性工作制，如接待员、销售人员等。

第二节 激励后果分析

一、组织公民行为

（一）组织公民行为的含义及结构维度

20 世纪 70 年代之前，泰勒的科学管理成为美国企业的主流管理思想。它强调“严格遵守官方规定的职责、纪律和制约”，员工的工作行为被限制于固定的职责范围之中，具有刻板狭窄的特点。20 世纪 70 年代之后，日新月异的科技进步将企业置于一个快速变化的环境之中，全球化的进程更增加了环境中的不确定因素。一方面，让员工拥有自主权来灵活应对多变的环境，使员工乐于对组织做一些超越正式职责要求但对组织有益的好事，成为企业发展的必需；另一方面，知识经济时代产生了越来越多的知识性企业，它们的竞争优势来自于获取和利用知识。但思想的交流和传播很难用职责任务定量地加以考察，知识员工的工作产出难以具体衡量，其工作角色难以被完全描述和定义，也亟需一个新的概念进行衡量。

1983 年，贝特曼（Bateman） 和奥根（Organ）提出了“组织公民行为”（Organizational Citizenship Behavior，简称 OCB）的概念。所谓组织公民行为是指员工自觉自愿地表现出来的、非直接或明显地、不被正式的报酬系统所认可的、能够从整体上提高组织效能的个体行为。从这个概念中，可以抽取出组织公民行为的几个重要特征：

第一，这些行为必须是组织成员自觉自愿表现出来的行为；

第二，组织公民行为是一种角色外的行为，即是一种未被组织正式规范或工作说明书规定的行为，但却是组织所需要的；

第三，组织公民行为不是由正式的奖惩系统来评定的行为，它的完成不会被组织奖赏，而员工不从事这些行为也不会为组织惩罚。

对于组织公民行为的结构维度，许多研究者都提出了自己的观点。虽然在划分时存在二维结构、三维结构、四维结构、五维结构和七维结构等的不同，但在组织公民行为的多维性上，已达成了共识。综观以往所有研究，得到认可较多的还是奥根所提出的组织公民行为五维结构，即：（1）利他行为，帮助别人完成与组织有关的任务和工作的自主行为，包括帮助处理或阻止工作中发生或即将发生的问题，鼓励在工作或个人职业发展方面失去信心的同事等，该维度被认为是公民行为的主要形式；（2）文明礼貌，对别人表示尊重的礼貌举动；（3）运动员精神，员工在非理想化的环境中毫无抱怨、坚守岗位的一种意愿行为；（4）责任意识，严肃认真、尽心尽责对待工作，将组织规则、制度和程序内化为个人要求，即使无人监督的情况下仍能一丝不苟地遵守规则；（5）公民美德，积极参加和自觉关心组织各项活动，把自己视为组织的一部分，为组织利益甚至不惜牺牲个人利益的行为。

在中国背景下对组织公民行为的结构维度进行研究发现：中国文化背景下的组织公民行为在利他行为、责任意识、公民美德三个维度上与西方相似，而保护公司资源和人际和睦两个新维度取代了西方背景下的运动员精神和文明礼貌。究其原因，前者是因为在中国集体主义的文化背景中，员工公私分明、不占组织便宜被认为是组织公民行为的一个重要表现；后者则是因为中国人“和为贵”的价值观发散性地波及到人际关系的各个领域，使得他们对人情、面子、人际和谐异常重视，因此维护人际和谐的行为在中国文化中被认为是重要的角色外组织促进行为。

（二）影响组织公民行为的因素

影响组织公民行为的因素有许多，主要包括工作满意感、公平知觉、组织承诺和领导支持等。

工作满意感是人们对工作的一种积极的情绪与情感性的反应，源于对工作职务或经历的评价（详细内容参见第十二章）。工作满意感越高，员工就会越多地表现出积极的情绪状态，从而驱动利他行为的产生。

从亚当斯公平理论的角度来看，员工与组织的关系是一种社会交换关系。在这种关系中，组织公民行为是一种投入因素，增加或减少组织公民行为是对公平或不公平的反应。由于组织公民行为是一种可自由决定的、正式角色之外的行为，因此在认知到的公平感发生变化时，改变组织公民行为是员工可以选择的一种最安全的策略。公平感增强，员工就会增加组织公民行为的投入，反之则会减少投入，甚至表现出利己行为。

组织承诺是指个体认同并卷入一个组织的强度，它反映了员工在不能获得直接回报的情况下对组织提供帮助的意愿，以及愿意长期保持组织成员身份的意愿（详细内容参见第十二章）。由于组织承诺是一种可以使人为了组织牺牲个人利益的工作态度，因此它被研究者们假设为组织公民行为的一个预测变量。研究发现，组织承诺对组织公民行为具有影响作用。

领导支持是指员工在多大程度上认为领导是重视自己的贡献并关心自己的福利的，借此来反应员工与领导之间的关系。高水平的领导支持使员工相信领导会在领导—下属交换关系（Leader—Member Exchange）中履行他的义务（如奖励员工），员工就会以一种强烈的义务感，做出更大的贡献回报领导和组织。研究表明，领导支持对工作绩效、组织公民行为和组织承诺都有显著的影响作用。

（三）组织公民行为的作用

奥根认为员工们长期一贯的组织公民行为可在组织运作过程中充当“润滑剂”，减少组织各个“部件”运行时的相互摩擦，从而提高组织的效能。其作用具体体现在以下几个方面：

第一，组织公民行为可以提高员工的生产力。员工可以通过相互帮助（利他）更快地提高自己的工作技能，更高效的工作方法可能因为员工的互助而在工作团队中传播开来；

第二，可以提高管理效能。如果员工积极参加公司的事务（公民道德），那么管理者就能够获得更有价值的建议，以及员工对管理措施的反馈，便于进一步提高管理效能；

第三，可以释放更多的资源，投入到生产性活动中。责任意识强的员工需要的外部监管比较少，管理者可以更放心地授权，节省出时间和精力投入到其他方面的管理中；

第四，能够有效协调团队成员和工作群体之间的活动。积极主动地参加公司活动有助于团队成员之间协调彼此的努力方向，进而提高团队的效率；

第五，能够创建良好的企业文化，增强组织对优秀人才的吸引力。利他行为可以提高士气、团队凝聚力，增强归属感，这些都能够帮助组织吸引和留住更好的员工；

第六，能够提高组织的稳定性。互助行为可以使员工同心协力共赴难关，保持稳定的工作成绩，责任意识也有助于员工保持持续高水平的工作业绩；

第七，能够提高组织适应环境变化的能力。组织公民感强的员工可以主动发现市场变化的信息，出主意想办法帮助组织适应环境。有运动员精神的员工愿意适应新的挑战，乐于学习新技能，适应新要求等。

二、工作投入

工作投入是心理学家和社会学家共同关心的一个问题。从组织的角度看，工作投入是激发员工动机的关键和建立商业市场竞争优势的基础。有效的激励可以增加员工的工作投入，使员工全力以赴地工作。从个体的角度看，对大多数人而言，工作活动消耗了生命的大部分时间，构成了生活质量最根本最重要的部分。人们可能愉快地、全神贯注地投入工作，也可能从心理和情感上疏远工作。因此，个体的整个生活质量可能在很大程度上受到其投入或疏离工作的影响。该部分就从工作投入的概念、结构、测量及其相关研究等方面作一分析。

（一）工作投入的概念

对工作投入的界定，学术界存在一定的分歧，总结起来主要有以下三种观点：

观点一：工作投入是个体在心理上对工作的认同程度。该概念下的工作投入可被看成

是“内在激励”的结果。当员工感知到通过个人努力而获得的高工作绩效可以使自尊需要得到满足时，个体就会在工作上产生投入。该定义从认知的层面对工作投入进行了界定：工作投入是个体内心对工作重要性的一种认知结果。

观点二：工作投入是个体投注精力、时间及兴趣于其工作上的程度。该概念不认为工作投入是员工发自内心的想法，员工可能对工作不很喜欢不很满意，但由于将高质量地完成工作视为自己的责任，因此也会投入自己的精力、时间及兴趣于工作之上。该定义从行为的层面界定了工作投入：工作投入是个人对工作的投入力度。

观点三：工作投入是工作表现对个体的重要性程度。当个体的工作表现受到组织的重视，同时个体知觉到组织可以满足自己目前的显著需求时，个体就会重视自己的工作表现，从而投入到工作中。该定义从情感的层面对工作投入进行界定：工作投入是工作表现对个体的重要性程度，表现为在工作情境中受到越多肯定时，个体就会越投入工作。

对于以上观点，一些研究者支持其中一种或几种，一些研究者则全面赞成。笔者认为，这三种观点从本质而言是不矛盾的，仅仅是看问题的角度不同。从认知、行为、情感三个层面共同界定才是最全面完善的。因而将工作投入定义为：由于个体在心理上对工作的认同，因而引发的重视个人工作表现的态度和积极主动参与工作的行为。

（二）工作投入的结构和测量

工作投入在概念上的复杂性，直接导致了测量的混乱。持单因素论的研究者坚持将工作投入看成“个体对工作的心理认同的一种总体的认知（或信念）状态”，因此试图直接测量个体自身对工作的认同感。持多因素论的研究者则将工作投入看成是一个建立在认知、活动和情感基础之上的综合概念。其结构包含四个维度：（1）工作作为一种主要的生活兴趣；（2）个人积极参与工作的程度；（3）工作表现与自尊的关联程度；（4）工作表现与自我概念的一致性。据此，编制了工作投入的多维度量表。我国研究者以中国员工为对象，编制量表进行考察后发现，我国员工的工作投入与西方有异有同。我国员工工作投入由四个维度组成：兴趣导向、心理认同、自我实现、积极参与。兴趣导向是指个体在目前工作中体会到了乐趣，并把工作看成自己的兴趣所在；心理认同是指个体心理上对目前工作的认同程度，对目前所从事的工作感到自豪；自我实现是指个体认为在工作中能实现自我价值，对未来工作充满期望；积极参与是指个体在工作中花费精力以及投身其中的程度，表现为在工作中追求完美。

（三）工作投入产生的原因

对于工作投入产生的原因，主要存在三种观点，即个体差异论、情境决定论和交互作用论。个体差异论认为工作投入是一种个体差异变量，它主要是个体在社会化过程中将传统的工作价值观内化的结果。因此，它与职业伦理道德和个人的责任感紧密相连，具有相当的稳定性，不会随具体工作环境特点的变化而变化。而情境决定论者则认为，工作投入是工作情境的函数。许多工作因素如组织的特点、工作的性质等都可能影响员工的工作投入程度。交互作用论者则综合了前两者的观点，认为不能简单地把工作投入视为人格变量或情境变量决定的，工作投入是人格特征和情境特征的函数，它既受组织条件的影响也受到早先社会化过程中习得的价值取向的影响。

按照交互作用论的观点，研究者经常考虑的影响工作投入的因素可分为以下几类：人格特征（如对职业伦理的认同、控制点、自尊、成长需要的强度、内部动机等）、工作特征（如工作的自主性、技能多样性、任务的特点、反馈、任务的意义、挑战性、任务复杂性等）、管理行为（如能否体谅员工、参与决策程度、交流的质量和数量等）、角色知觉（包括角色含糊和角色冲突）。此外，还有一些可能与工作投入相关的因素，如人口统计学变量（包括年龄、性别、受教育程度、婚姻状况、任职年限等）、职业承诺、组织类型（公有组织或私有组织）等。研究发现，人格变量中，对职业伦理的认同、控制点、自尊三个变量与工作投入有显著相关；情境变量中，技能多样性、任务的特点、反馈、任务的意义、挑战性、任务复杂性都与工作投入有显著相关；管理行为变量中，参与和体谅两变量影响效果显著。可见，人格变量和情境变量都是工作投入的前提条件。另外，研究表明，人口统计学变量与工作投入之间没有关系，因此不能用年龄、性别、受教育程度、婚姻状况、任职年限等因素来预测工作投入。

（四）工作投入的后果

工作投入的结果变量中，工作满意度与工作投入的关系最强。究其原因，可能是因为工作投入和工作满意度都是对一种具体工作的反应，只不过前者是一种认知状态，后者是指喜欢某一种工作的情感状态，其指向的内容是相同的。工作投入与离职和旷工关系不明显，对绩效的预测能力也较弱，其原因可能是一些未知的心理和行为变量（如动机、情感、创造力、合作、努力程度等）在其中起到调节作用。另有研究还表明，工作投入与内部需要的关系要高于与外部需要的关系，这也再一次证明：有趣的、具有挑战性的工作对培养工作投入的重要性。

本章小结：

1．薪酬是员工因向所在组织提供劳动或完成任务而获得的各种形式的酬劳或酬谢。它是劳动力价格的支付形式，在市场经济环境下，同时又是人力资本竞争的价格表现。薪酬按照存在的形态不同，可分为经济性薪酬和非经济性薪酬。

2．宽带薪酬是对传统的带有大量等级层次的垂直型薪酬结构的一种改进或替代，是指对多个薪酬等级以及薪酬变动范围进行重新组合，从而变成只有相当少数的薪酬等级以及相应的较宽薪酬变动范围。

3．员工持股计划是指企业内部员工以个人出资（大都指未来劳动的收益）的方式，或贷款融资的方式，来持有本企业的一部分特殊股票，并委托员工持股会集中管理的一种产权组织形式。员工持股计划本质上属于一种股票奖励计划，员工可借此渠道参与企业经营管理和分享利润。

4．工作设计是指对工作完成的方式以及某种特定工作所要求完成的任务进行界定的过程，包括岗位轮换、岗位扩展、工作丰富化、弹性工作制等形式。

5．组织公民行为是指员工自觉自愿地表现出来的、非直接或明显地、不被正式的报酬

系统所认可的、能够从整体上提高组织效能的个体行为。

6．工作投入是指，由于个体在心理上对目前工作的认同，因而引发的重视个人工作表现，并主动参与工作的行为。

本章思考题：

1．简述薪酬的概念与分类。
2．运用薪酬对员工进行激励时，应注意哪些问题？
3．简述目前出现的新的薪酬激励形式。
4．简述组织公民行为的概念、维度划分和作用。
5．简述工作投入的概念、维度划分和相关理论。

案例分析：

从 CENT 离职看 GC 的薪资体系

GC 是一家由原国家事业单位转型为企业的公司。现有员工 200 余人，离退休人员 150 余人——均为原事业单位工作或离退休人员。在经历了计划经济向市场经济转变的阵痛后，GC 充分利用现有资源，并逐渐融入到了激烈的市场竞争中。

但由于各种条件（包括宏观环境和微观环境，当然这里主要是指宏观环境）的限制，特别是经费来源方面原来一直依赖中央财政直接拨款，因此 GC 的改革步伐非常小——这种改革的速度远远落后于整个社会的变革速度，而这些变革中尤显滞后的方面便是薪资方面的大锅饭和用人方面的论资排辈。

2003 年末的一天，一位被各级领导看好的办公室员工 CENT 提出要离开公司。当 CENT 向其上级 S 提出要走并希望立即得到批准时，S 半晌没回过神来，她不相信 CENT 会选择离开——至少不应该这么快。

S 迅速地回忆了一遍 CENT 的成长历程：从一线干起，历经数个工作岗位，一直表现优秀；为公司争得了好多荣誉；曾被提名为 2000 年度省优秀共青团员；现在已经是共产党员；任公司团支部书记，并分管青年工作；函授本科马上就要毕业了……而且，他才 22 岁！这么一个自己看着成长起来，工作认真、学习上进、生活严谨的年青人，为什么会突然离职？

带着这个问题，S 与 CENT 进行了坦诚的交流。CENT 对这位曾全力培养过自己的领导没有任何隐瞒，他向 S 陈述了自己决定离职的三点原因：

第一，公司所能提供的环境已不能满足他对自己高标准成长锻炼的需求。

第二，公司所给予的待遇与自己所做出的贡献不成正比，而且即使今后发展顺利，其收入增长速度仍远远跟不上自身能力与素质的成长速度。

第三，现在这个社会已不存在终生的职业，只存在终生的能力。

S 对 CENT 提出的一、三两点并没有思考太多，因为一个组织的条件总是有限的。当员工在自己的规划中，对公司的要求已超出了公司所能做到的极限，那么对于一个组织特别是对于一个体制还没有完全转变过来的组织而言是无力改变这种现实的。但关于员工的

工资，似乎有一定的调整空间。

由于公司是从原事业单位转型而来，工资等方面仍大部分沿用了原体系，该体系大致情况如下：

（一）以岗定薪

薪资标准主要以该员工所在岗位的重要性和工作量大小来衡量。岗位是确定员工工资的基础。

（二）将员工工作年限考虑到整个体系中

员工工资调整是定时进行的，只要员工在工作中没有犯大的错误，就可以享受加薪待遇，对表现特别突出的可超级或提前晋升工资档次。

薪资基本形式如下表（未按实际数字列出）：

岗位 \ 数额 \ 档次		一 岗位工作 2 年以下	二 岗位工作 2～3 年	三 岗位工作 4～5 年	四 岗位工作 6～7 年	五 岗位工作 8～9 年	
行政	一级职员	1200	1300	1400	1550	1700	
	二级职员	800	850	900	980	1060	
	三级职员	600	630	660	700	740	
技术人员	高级	1200	1300	1400	1550	1700	
	中级	1000	1060	1120	1200	1280	
	初级	800	850	900	980	1060	
	技术员	700	740	780	830	880	
工人	高级工	900	950	1000	1060	1120	
	中级工	700	740	780	830	880	
	初级工	500	530	560	600	640	
	普工	350	370	390	420	450	

从实际操作中可以看出，该薪资体系存在以下问题：

（1）薪资与学历无联系，这导致了员工整体素质不高。

（2）各岗位薪资无明显差异。最可以说明问题的事实是：公司守门人（高级工五档 1120 元）与公司一把手（一级职员三档 1400 元）的工资差距仅为 280 元。这种情况导致了员工进取心普遍偏低——投入与产出严重失去平衡。

（3）如果工作岗位不发生变化，无论员工的表现有多优秀、员工自身素质有多高，均无法在薪资中得到体现。

当然，这种薪资体系亦有其优势，主要表现为：

（1）由于严格执行薪资政策，同岗位员工薪资基本上没有什么差距，绝大部分员工情绪稳定。

（2）这种薪资制度是透明的，员工一旦选择进入公司，即表示接受了这种制度，抱怨的情况不多见。

鉴于以上情况，S 产生了如下想法:

（1）可否将学历因素考虑到薪资体系中去？这不仅可以让正在进取中的员工清楚地看到学历方面的差距，还可以达到提高全员素质的目的，更可体现公司对知识的尊重，从而吸引和留住组织需要的优秀（高知）人才。

（2）可否拉大档差？在本着对社会和基层员工负责的态度的基础上制订公司起薪标准，重视员工成长。

（3）可否对特殊人才实行谈判工资（该工资不在上述工资框架内）？

（4）可否将员工调薪的依据从单纯的工作年限中剥离出来，主要运用考核的手段定期或不定期地对表现优秀的员工提名晋升，将工资导向从资历转变到工作贡献上来（对服务年限以补贴形式单独进行补偿）？或者考虑将原薪资中本身就存在的 30%的活动工资与考核成绩真正联系起来——打消员工将这部分工资作为固定收入的一部分的守旧思想？

（5）可否根据工作地点不同，按当地政府的最低工资标准为依据适当调整实际工资或给予补贴，以平衡跨地区同岗位间实际可比收入？

（6）如何才能在组织承受能力以内让员工得到（或感受到）更多的非工资收入？

(资料来源：中国人力资源网)

问题：

1. 你认为 S 解决 CENT 离职问题的思路正确吗?为什么?
2. 你认为 S 改进薪酬的想法合理可行吗?还有哪些改善余地?



第三篇　认知管理

第七章　组织中的知觉

知觉是对感觉信息输入进行主动处理并赋予其含义的过程。在组织管理过程中，管理的对象是人，管理是一个与人交往、对人产生认识，并与人发生相互作用的过程。管理双方在有限的时间内利用可用信息，对个体的需要、动机以及个性特征等做出迅速有效的判断，进而采取适当应对措施的过程就是组织知觉加工过程。个体就是通过这一过程对组织中的个体，以及组织本身进行认识的。本章从知觉概念入手，对组织知觉中的障碍、归因等问题进行论述，并对实际应用提出了建议。

第一节　组织中的社会知觉

知觉是指个体选择、组织和解释感觉器官收集的信息，从而理解周围世界的过程。知觉的对象包括人和物两部分，对人的知觉称为社会知觉，是管理心理学研究的主要内容。在工作生活中，每个人每天都接触包括上级、同事、下属、家人、朋友、客户在内的大量不同的人。对所获得的信息进行整合和解释，形成关于自己或他人的印象，加深对自己或他人的了解，就是社会知觉过程。其中对自己的体貌、个性、能力等方面特征的知觉过程称为自我知觉，对他人各方面特征的知觉称为人际知觉。

一、组织中社会知觉的种类

（一）自我知觉

自我知觉的主要成分是自我概念和自尊。自我概念是个体关于自己的信念的集合；自尊则是个体对自己的评价及其带来的情绪体验。个体对自己的认识有多种来源，包括社会化、角色认知、对自己行为的观察、他人的反馈、与他人的比较、经历过的重要事件等。

自我概念（self-concept）是人格结构的核心部分，它是个人对自己所有方面（包括生理、心理等）的知觉和主观认识，这种认识是个体在社会化过程中逐渐形成与发展起来的，是一个关于自己及自己与周围环境关系的多方面的、多层次的认知评价体系。自我概念由四个部分组成：物质自我、社会自我、精神自我和纯自我。其中物质自我是基础，社会自我高于物质自我，精神自我在最高层。

自尊是个体对自己做出的评价，即个体的自我价值感。自尊在很大程度上受到情境的影响。成功会提高自尊，而失败会降低自尊。高自尊的个体对自己持正面情感，认为自己既有优点又有缺点，但优点比缺点更重要。低自尊的个体则对自己的看法比较消极，他们

更多地被他人对自己的看法所影响，行为上倾向于趋近给自己正面反馈的个体，回避给自己负面反馈的个体。多数情况下，高自尊个体更乐观、心理更健康，而低自尊的个体可能抑郁、焦虑和有某些行为问题。一个人的自尊影响许多其他态度，并且对组织中的行为有重要意义。高自尊的人们工作更出色，并且对自己的工作更满意。求职时，他们希望获得更高的职位。研究表明，一个由高自尊个体组成的工作小组比平均自尊低的工作小组成功的可能性更大。当然，自尊过高的人处于充满压力的情境中时，可能会不适当地自夸，这可能造成他人的负面评价。

由于总的来说高自尊被认为是一种正面的特点，所以管理者应该注意保护下属的自尊，给员工适当的挑战和获得成功的机会，从而达到激励目的。需要注意的是，单纯的表扬、鼓励或吹捧并不能真正提高自尊，自尊的主要来源是成功体验，即通过努力取得成功后，自尊才会有真正的提高（Bandura, 1997, 2000）。一个需要考虑的问题是：到底大多数人对自己的评价是偏低还是偏高呢？以罗杰斯为代表的人本主义心理学家认为，大多数人对自己的评价偏低。但是，与此观点相反，很多社会心理学研究发现，多数人的自尊都偏高，表现出用对自己有利的方式知觉自己的倾向，即自利性偏差。自利性偏差主要有4种表现：

（1）人们对积极和消极事件的解释方式不同。他们往往把成功归结为自己的才能和努力，而把失败归结为运气差或问题本身的难度等外在因素。研究者在婚姻、企业、谈判等情境中都发现了这类自利性偏差，它会导致婚姻不和、员工不满和谈判时的僵持局面。在组织中，当公司利润增加时，CEO们往往会把这一成功归功于自己的管理能力，而当利润下降时，他们又会归咎于经济不景气、员工缺乏能力或工作不够努力等。企业中的员工也是如此，他们往往将失败归咎于工作负担过重、同事不易相处、目标不明确等。而且，只有当人们得到的奖金比大多数员工都高时，他们才认为是公平的。

（2）每个人都认为自己高于平均水平。当人们把自己与别人进行比较时，往往认为自己符合社会赞许的方面多于平均水平。当问及自己的职业能力时，90%的商业经理认为自己的业绩优于同级经理的平均水平。澳大利亚的一项研究表明，86%的人把自己的工作成绩评价为高于平均水平，只有1%的人评价自己低于平均水平。多数外科医生认为自己病人的死亡率低于平均水平（Gawande, 2002）。

（3）对未来的生活事件盲目乐观。大学生往往认为自己远比其他同学更可能找到好工作，获得更高薪水和拥有自己的房子，而诸如酗酒、40岁前突发心脏病等消极经历则不会发生在自己身上。尽管乐观可以提高自我效能、促进健康和幸福感，但是过分乐观会导致不去采取预防危险发生的措施，事前不做充分的准备工作，赌博时即使不断输钱仍然坚持等。因而，盲目乐观往往会造成最后的失败。

（4）人们容易高估自己的观点和缺点的普遍性，同时低估自己能力和品德的普遍性。例如说谎的人往往认为别人也不诚实，逃税或吸烟的人认为大多数人都与他一样（Sagarin & others, 1998）。相反，当人们工作做得不错或取得成功时，则认为自己优于他人。

自尊偏高所造成的自利性偏差有一定的适应意义。高自尊的人通常自我感觉良好、更能应付日常生活压力、心理更健康。但是，这种偏差也会带来不利影响。研究表明，自利性偏差在一定程度上会毁掉整个群体，这一点对组织来说非常致命。一些心理学家对群体

成员的自利性偏差进行了一系列实验。他们让被试共同完成某些任务，然后得知任务完成得很棒或很糟糕。结果发现，成功组的成员宣称对他们组的成功做出了贡献，多数人认为自己比群体中其他成员的贡献要多，而失败组则刚好相反。如果一个组织中的成员都认为，自己做出了超出平均水平的贡献，但是又没有获得相对应的报酬和未得到充分赏识，那么很可能会产生不和谐和嫉妒。在这样的组织中，当宣布加薪时，总有一半的人得到平均或低于平均水平的加薪，那么他们很可能会认为自己是不公平的受害者。

（二）人际知觉

人际知觉是一个“以己度人”的过程，即根据自己的经验、知识等对他人的信息进行解释。几乎所有的管理活动都依赖于人际知觉。在评价业绩时，管理者利用他们对员工行为的知觉作为评价的基础；在招聘中，初次面谈的前 4、5 分钟，招聘者通常已经根据他或她对应聘者的知觉，做出了接纳或拒绝的决定。

主要有三类因素影响个体对他人的知觉：知觉者的特点、目标个体的特点，以及情境的特点。

1．知觉者的特点

知觉者的几个特点会影响人际知觉。首先是对被知觉目标的熟悉性。当与被知觉对象熟悉时，个体可以根据多个观察来源建立对他或她的印象。如果在这些观察中收集的信息是准确的，那么就可以形成对这一个体的准确知觉。但是，熟悉并不总是意味着准确。熟悉度过高往往使得认知者倾向于滤掉那些与已有信念不符合的信息，从而扭曲了事实的真相。这在业绩评估时格外危险，因为业绩评估的对象通常都是十分熟悉的人。管理者需要注意避免先入为主的倾向。

知觉者的态度也影响人际知觉。假设你在为组织中一个重要的职位访谈应聘者，这个职位需要与供应商协商合同，从事这一职位的多数是男性。此种情况下，你可能会感觉女性不擅长在谈判中坚持自己的立场。这种态度无疑会影响你对女性应聘者的知觉。

情绪也对知觉他人的方式有影响。个体在高兴时与沮丧时思考问题的方式是不一样的，并且倾向于记住与情绪状态一致的信息。当处在积极情绪状态时，个体倾向于形成对他人的积极印象。当处在消极情绪中时，倾向于对他人形成消极的评价和印象。

另一个影响人际知觉的因素是知觉者的自我概念。持有正面自我概念的个体倾向于注意他人的正面属性，而负面自我概念会导致知觉者看到他人的负面特点。对自我的更多理解可以使个体获得对他人的准确知觉。

个体的认知结构，即思考问题的方式，也会影响人际知觉。一些人倾向于获得知觉对象身体特点的信息，例如身高、体重和外表等。另一些人则倾向于更多关注对方的人格特点。认知复杂性允许个体知觉他人的多个特点，而不是只关注少数特质。

2．知觉对象的特点

知觉对象的特点也影响人际知觉。身体外表例如身高、体重、大致年龄、种族、性别、衣着等在人际知觉中起重要作用。研究表明，那些与规范相反的、新的或不寻常的外表特征，以及新异的行为较容易引起知觉者的注意。说话大声的、衣着古怪的、身材和长相特

殊的以及过度活跃的个体都很容易成为知觉的中心。此外，外表吸引力通常会影响他人形成的印象。尽管面试官可能会加以否认，但是吸引力和外表修饰确实会影响招聘访谈中的第一印象。这一规律帮助解释了为什么有吸引力的人通常获得了更有声望的工作，并且赚钱也更多。一些研究考察了被试的外表吸引力水平与收入之间的关系，发现吸引力得分每增加一个单位，被试的年收入可增加 2000 美元左右（Roszell & others, 1990; Frieze & others, 1991）。

知觉对象言语交流方面的特点也影响人际知觉的形成。他们谈论的主题、他们的音调、他们的口音都会作为知觉者做出判断的信息依据。

知觉对象非言语交流方面的特点也传递着大量的信息。有时人们会尽量隐藏自己的真实信息或感受，甚至会撒谎。这时候，知觉者可以观察知觉对象的目光接触、面部表情、身体动作和姿势，从而建立起印象。这些行为相对来说较难控制，因此传递的信息较为真实可靠。一些非言语的信号在不同文化背景下有不同的含义，但研究发现面部表情有普遍的含义，来自不同文化背景的个体以同样的方式识别和解读表情。

知觉者还会观察知觉对象的行为，从而推测其意图。当看到老板出现在办公室门口时，员工往往会不自觉地推断他此行的意图：是要安排更多的工作，还是打算祝贺最近取得的成功。无论如何，知觉者依据对方行为对对方意图的解释影响其看待知觉对象的方式。

3．情境的特点

知觉者和知觉对象相互交往的情境也影响形成的印象。交往发生的社会情境对人际知觉有重要影响。在教授的办公室遇到他，还是在一家饭店相遇，两种情况下你建立印象的方式有很大不同。在日本，社会情境显得尤其重要。在工作时间之外或午餐时谈论生意是一种避讳。如果你在这些时间试图谈生意，将被知觉为粗鲁的。

情境线索的强度也影响人际知觉。一些情境提供了关于什么是适当行为的强有力的线索。在这种情境下，我们会假定个体的行为可以被情境所解释，但却不能反映个体的性格倾向，这叫做人际知觉的折扣原则。例如，汽车展销会上热情又有魅力的汽车推销员，他询问顾客的工作和娱乐，并且对顾客的汽车品味很感兴趣，表现得周到又热情。你会认为这种行为反映了销售员性格？可能不会，由于情境的影响，你会将“试图卖车”知觉为该销售员的真实意图，而不会认为“热情”、“有魅力”是他个性的表达。

二、社会知觉的障碍

尽管人们希望形成对他人的准确认识，但有很多因素造成形成的印象并不准确。研究者已经证实，许多知觉偏差和倾向会干扰人们对他人的准确判断。分析这些偏差，有助于找到避免它们的方法。

（一）第一印象

通常来说，个体对某个人的判断并仅仅依据当前获得的信息，在很大程度上依赖于最初对他形成的判断，即第一印象。一系列印象形成的研究发现，最先描述的特征对人们印

象形成的影响巨大。当第一印象影响过于强烈，严重阻碍了准确的印象形成时，被称为第一印象错误。

在组织中，一项关键的管理任务就是判断他人业绩，第一印象错误往往会给绩效评价带来严重后果。当一名下属的业绩有进步，需要识别和奖励时，如果当前的评估只是根据糟糕的第一印象，那么管理者可能意识不到他的进步。同样，当一个最初给领导留下良好印象的人表现糟糕时，由于第一印象的作用，管理者也可能面对证实该下属业绩下滑的证据而仍视而不见。一项研究表明，第一印象错误也可能以微妙的形式存在。该研究的被试是公司的面试官，他们事先评估应聘者的申请表格和测验分数。研究者发现，被试根据这两项标准对应聘者作出的评价越高，他们在访谈过程中对待应聘者的态度越正面。对访谈录像的分析发现，最初给人形成正面印象的应聘者得到更好的对待，被试与他们交谈的方式更愉快、互动更多，并且更多地告知他们关于公司的正面信息。也就是说，招聘者并不是利用访谈来获得更多的无偏见的信息，而是用访谈来确认他们根据表格和分数形成的第一印象。

（二）晕轮效应

人们经常认为，聪明的人会努力工作，蠢笨的人比较懒惰，这种知觉偏差被称为晕轮效应。当个体建立了对某知觉对象某方面的正面印象时，就倾向于认为该人的所有方面都是好的。类似地，当形成对某人某方面的负面印象时，就倾向于认为该人的所有方面都是不好的。这两种现象都属于晕轮效应。常见的晕轮效应是根据个体的外表特征来推断他的其他特征，例如，长相漂亮的人也被认为具有和蔼可亲、善良和受欢迎等特点。在组织中，当管理者使用正式的业绩评估表格评价下属时，晕轮效应很容易出现。在这种情况下，当管理者将某个雇员评价为在某些维度上表现良好时，很可能会认为该雇员在其他方面也很好，从而在别的维度上的评分也较高。晕轮效应很可能是造成雇员在不同维度上的分数呈现高相关的原因。当出现这种现象时，评价的结果是不可靠的。

（三）选择性知觉

知觉的重要特征是具有选择性。我们并不是不加区别地接受所有眼前的事物，而是受到动机和认知等造成的预期的影响，只集中于环境中的某些方面，忽略其他方面。由于人们处在非常复杂的世界中，周围有过多的刺激，知觉的选择性可以使个体用有限的资源去关注那些对自己重要的信息，因而是具有适应意义的。但是，选择性知觉也可能会造成我们的知觉偏差。

研究证实，选择性知觉可以在组织中起作用。早期的研究已经发现，当要求企业管理人员阅读一份某工厂的案例时，销售经理认为案例中最重要的问题是销售，生产经理则认为最重要的问题是生产，其他人也各自认为自己负责的领域是案例中最重要的。另一项研究调查了来自很多组织的大量高级管理人员，要求他们回答组织的哪些方面对效率至关重要。结果发现，经理们对符合他们背景的方面更关注。例如，销售或市场背景出身的经理认为公司的产品和服务线路的变化最重要。研发背景出身的经理则认为产品的设计最重要。

也就是说，经理们受到选择性知觉的影响，他们更多关注那些符合他们背景经历的商业环境。这也是造成人们对同一情境产生不同知觉的重要原因。

（四）自我实现的预言

信念会影响感觉和行动，并最终变成现实。罗伯特 • 默顿（1948）将其定义为自我实现的预言。默顿指出，如果人们相信银行即将倒闭，纷纷排队去提款，他们错误的信念最终会制造出现实；同样，如果人们相信股票将上涨，他们的信念和由此导致的行为可能会造成股票真的上涨。

自我实现的预言最著名的证据来自罗森塔尔和雅格布森（Rosenthal & Jacobson ,1968）的研究。在他们的研究中，小学教师得到了关于学生的一些信息，这些信息暗示 20%的学生将在未来的学业上取得重大进展。结果发现，在 8 个月后的智商测试中，这 20%的学生的成绩确实比其他学生进步快。而实际上，这 20%的学生最初是随机挑出的，与其他学生并无不同。当老师相信他们有突出能力时，最终这一信念就变成了现实。这一现象也叫皮格马利翁效应[①]。

自我实现的预言在工作环境、法庭、伴侣之间、面试等情境中都存在。以色列国防军告诉一组海军学校学员，他们很可能不会感觉到晕船，即使晕船，也不会影响他们的表现。结果发现自我实现的预言起了作用，这组学员表现比其他学员好，晕船的人也更少。在组织中，管理者对个体的预期既影响管理者本人的行为，又影响预期对象的反应。当管理者认为某个雇员有升职的潜力时，就可能花很多时间指导他，给他提供更多的挑战任务，从而培养这一个体走向成功。由于对个体的高预期可以变成现实，管理者可以利用皮格马力翁效应来提高组织的生产力。研究表明，当一名经理对一个群体持有高的期望时，整个群体都表现更好。

（五）刻板印象

刻板印象是人们对某个特定群体的成员共有的特征所持有的固定观点。刻板印象的存在形式通常是：“来自 X 群体的人拥有 Y 属性”。例如，认为戴眼镜的人勤奋、害羞、不活跃，老年人不能学习新知识，女职员不善于坚持自己的观点等。刻板印象意味着，由于认定了特定群体的所有成员具有类似的特征和行为，因此根据某个人属于某个群体成员的身份，推断这个人的特点。刻板印象多数情况下都是负面的。

刻板印象常常也有一些根据，并且也有一定的益处。例如，它可以节省人的认知资源。研究表明，人们在思考他人时，倾向于尽量降低所需要的认知努力的程度。把某个人归于某一群体，假定这一群体的人有类似的特点和行为，据此认为这个人具有群体已知的特点，这样可以节省详细了解这个人所需要的时间和精力，是一种有效的捷径。而且，我们每天都要接触许多人，了解每个人的每件事情是不可能做到的。因此，个体往往依赖可以利用的信息，如年龄、性别、种族、工作类型等，作为组织知觉的方式。

[①] 皮格马利翁，古希勒雕塑家，传说他钟情于自己雕塑的女神像，后来这个女神像果真变成真人与其结合。

但是，这种对某个群体的刻板印象往往并不准确，过于泛化，忽略了个体之间的巨大差异。刻板印象使得知觉者仅仅根据对象所属的类别过早地下判断，而不去搜集和了解更多的信息，即便形成了不准确的印象，也很难有机会被检验和改变。如果一个人力资源经理认为某个群体的成员是懒惰的，他可能会有意地避免雇用或晋升属于那个群体的个体。这个经理可能坚持相信他收集了所有必需的信息，做出了很好的判断。但是，尽管没有觉察到，他所持有的刻板印象已经影响了他的判断。造成的结果是，某些个体的命运已经早已被决定了，并不是因为他的行为，而是由于他属于某个特定群体。而且，由于上述自我实现预言的作用，被贴上标签的群体成员往往会表现出与上司刻板印象一致的行为。例如，一个典型的刻板印象认为记忆随着年龄的增长而衰退。研究发现，接受这一观点的老人比否认这一观点的老人更多地表现出记忆衰退。在工作环境中，性别刻板印象往往会影响管理者对女性工作绩效的评估。研究发现，当评估者为男性时和当女性从事与男性有关的活动时，女性职员往往会受到歧视。在多文化的工作团队中，人们经常对外国同事持刻板印象，而不是在了解他们之后才形成印象。来自不发达国家的团队成员通常被认为知识不够渊博，仅仅是因为他们的祖国经济或技术不够发达。这些刻板印象会降低工作团队的生产力，还会导致低落的组织士气。

三、克服社会知觉的障碍

知觉偏差的存在并不意味着人们有意做出不准确的判断，而是由于人们不能对信息进行全面的加工。下面一些建议有助于改进社会知觉的准确性。

（一）识别和面对刻板印象

必须承认，大多数人都持有并依赖于刻板印象，特别是涉及到新认识的人时，这一点尤为明显。尽管这一倾向很自然，但它很可能会造成错误的知觉。为此，最好能够识别我们所持的刻板印象。这样可以帮助你意识到它们的存在，并尽量减少他们对行为的影响。

（二）根据客观因素评价他人

用于评价他人的信息越客观，产生知觉歪曲的可能性就越小。

（三）避免匆忙做判断

即使对他人了解很少，人们也往往倾向于匆忙得出关于他人的结论。花些时间深入了解别人，而不是告诉自己你已经知道了所有有关他的信息。

第二节　组织中的归因

人们不仅想知道他人是如何行动的，还想知道他们为什么如此行动。工厂的生产率下

降，是由于工人变懒了，还是由于生产设备落后了?销售员称赞我的穿着，是因为他由衷地喜爱，还是另有所图?对自己和他人行为背后的原因进行解释的过程被称为归因。

一、归因的基本原理

归因理论描述了我们怎样解释自己和他人的行为（Heider, 1958）。归因理论有一些一般的假设，通常认为人们试图将个体的行为归结为内部原因（也叫个人归因）或外部原因（也叫情境归因）。假设一名学生在一次测验中的成绩很好，当对其进行归因时，如果把他的成绩归结为能力或努力，即为内部归因；相反，如果认为这次测验很容易，或者是他运气好，即把成绩归因于任务难度或运气，即为外部归因。人们在组织情境下，也经常进行类似的归因。老板解雇了一名职员，对于该职员而言，如果认为是由于自己违反了公司的制度，就是内部归因；如果认为是由于老板的冷酷无情，则是外部归因。归因方式有一定的个体差异。研究表明，成功导向的个体倾向于将成功归因于能力，失败归因于缺乏努力；失败导向的个体倾向于将失败归因为缺乏能力，并且由于这种归因模式，可能发展出无助感，弱化未来成功的希望和信心，甚至造成抑郁。那么，人们通常是如何做出归因的呢？对此有两种有影响的理论。

（一）对应推断论

在组织中，人们经常需要了解某人是哪种人：对手是否是一个强硬的谈判者，同事是否好相处等。要了解他人，需要观察他或她的行为，然后从行为中推断其性格。所谓对应推断就是判断一个人表现出来的行为是否与其内在性格或真实意图相对应的过程。

这种推断表面上看很简单。例如，办公桌很混乱的人被认为是马虎的，在电梯口摔倒的人被认为很笨拙。但是，办公桌的混乱可能是由于某个同事到处翻寻一份文件的结果，而摔倒是由于地板很滑，任何灵巧的人都可能摔倒。也就是说，任何表现出来的行为都有很多可能的原因。另一方面，人们经常有意掩饰某些性格特点和真实的意图，特别是当这些特点和意图与当时环境的要求或社会期望相悖时。例如，毛脚女婿第一次登老丈人家的门可能表现得很规矩、很本分；申请销售职位的个体在面试时可能表现得很健谈、很外向，即使他的性格是内向的。因此，简单地做出对应推断很可能会犯错误。

那么，怎样才能做出准确的对应推断呢？或者说，在什么条件下才能认为一个人的行为反映了他特有的内在属性或与他的真实素质相对应呢？心理学家琼斯和戴维斯（Jones & Davis）对此问题进行了长期的研究，提出了两个基本的原则：（1）不寻常结果原则：当活动者的行为具有一些相当独特的或不寻常的结果时，知觉者可以作出对应推断，即：将这种行为判断为与活动者的内在倾向相一致。一个人行为的结果越不寻常，越能够使他与别人区别开来，反映他的内在品质。例如，把某人撞倒会产生与其他可能的行为（如婉言谢绝、对某人吵嚷）不同的伤害性后果，从而体现出该人鲁莽的个性特点。再如，假设你的朋友接受了一项新工作。你得知这个职位薪水很高、工作很有趣、并且工作地点很诱人。这时，你难以判断对你的朋友来说什么是重要的，因为任何人都会接受这一工作。但是，

假设这项工作薪水不高，工作任务繁重枯燥，但工作地点是一个风景优美的城市。这时，你可以很容易地推测什么对你的朋友来说最重要了。一种行为不寻常的结果越多、越独特，它就越可能被判断为与活动者的内在倾向和品质相对应。（2）低于社会需求性原则：知觉者关于其他人在同样情境中将会做什么的信念也直接影响对应推断。当一种行为与社会上人们所期望和赞许的行为常模不一致时，人们就会将这种行为归因于活动者的内在属性。例如，演员浓妆艳抹是符合人们的期望的，由此难以推断他（她）是否具有喜欢化妆这种特点，而一般人若过分追求化妆，则能够提供关于他（她）的特点的信息。

（二）因果推断论

关于人们是如何根据观察到的结果来确定其原因的，归因理论的创始人海德（F.Heider,1958）曾经提出过一项著名的原则——协变原则，即看一看哪些因素与这种结果是协同变化的。后来，凯利（H.Kelley,1972）接受了这个原则，并给出了一个完整的定义：“如果一种结果存在，某种条件存在，这种结果消失，这种条件也消失，那么这种结果就归因于这种条件。”

在协变原则指导下，凯利提出了一种详尽的因果推断模型。他首先区分出了两类归因问题：一类是能够在多次观察同类行为或事件情况下的归因，称为多线索归因；另一类则是依据一次观察就做出归因的情况，称为单线索归因。凯利认为，人们在对特定的社会行为进行归因时，可供选择的原因不外乎有三类：（1）活动者个人方面的原因：诸如他的心境、个性、态度等；（2）环境方面的原因：诸如环境气氛、某种外界压力等；（3）刺激对象方面的原因：诸如活动对象的特点等。对一种行为结果究竟作出哪种归因呢？这取决于下列三种行为信息：第一种信息称为“一致性信息”（Consensus information），即看一看活动者的行为和多数人的行为是否一致，更具体地说，就是看一看究竟有多少人像活动者那样做了。例如，所有走相同路线上班的员工都迟到了，则迟到行为的一致性就高。从归因的观点来看，如果一致性高，我们对迟到行为作出外部归因。如果走相同路线的其他员工都准时到达了，只有一个员工迟到了，则该员工迟到行为的一致性低，可认为该员工迟到的原因来自于内部。第二种信息称为“区别性信息”（Distinctiveness information），即看一看活动者只对当前的刺激对象产生了这种行为，还是对许多不同的对象都产生了同样的行为。例如，一名员工今天迟到了，我们就要看一看他是否在其他事情上也表现得自由散漫、违反规章制度。如果行为的区分性低，则可以对其迟到行为作内部归因；如果行为的区分性高，则迟到的原因很可能来自外部。第三种信息是“连贯性信息”（Consistence information），即看一看活动者对当前的刺激对象是否一贯地都产生同样的行为。例如，如果一名员工并不总是上班迟到，她有 7 个月从未迟到过，则表明这是一次特例，行为的一贯性较低；而如果她每周都迟到两三次，则说明行为的一贯性高。行为的一贯性越高，观察者越倾向于对其作内部归因。凯利认为这三个方面的信息构成一个协变的立体框架，可以依据这三方面信息的协同变化将人的行为分别归因于行动者、客观刺激物或情境，如表 7.1 所示。

表 7.1　三种行为信息的协变与归因

行为信息			归因类型
区别性	一贯性	一致性	
低	高	低	活动者
高	高	高	刺激对象
高	低	低	环境

二、归因偏差

人们在推断自己和他人行为的原因时，并不总是客观的，常常出现错误或偏差。常见的两种归因偏差是基本归因偏差和自利性偏差。

（一）基本归因偏差

研究者发现，人们在归因时存在一种普遍的倾向，即当解释他人的行为时，往往会低估环境的影响，而高估个人特质和态度的影响。也就是说，即使存在明显的情境原因，人们仍然倾向于认为他人的行为起因是其内在的性格。由于这一偏差非常普遍，所以社会心理学家称之为基本归因偏差。例如，当发现一名员工迟到时，人们更多地将其归因于他的懒散，而不是交通堵塞；当发现一名职员把文件掉到地上，接着又打翻了咖啡时，人们倾向于认为这个人杂乱无章和笨拙，而不是考虑文件太滑或咖啡杯子太热。

之所以会出现基本归因错误，原因之一是用特质来解释一个人的行动很容易，而识别影响其行为的复杂情境因素则比较困难。当观察他人的行为时，人们倾向于注意他的行为本身，而忽略行为发生的情境。因而，与情境原因相比，内部原因更容易受到注意、更突出。

在组织中，基本归因偏差常常让我们过早推定别人的责任，而不考虑可能的外部原因。从而造成对他人的不准确判断。当员工业绩较差时，基本归因偏差会导致管理者完全忽略可能造成这一结果的不可控原因，过多地责备员工。要想避免这一偏差，就应该设身处地考虑别人的处境，即进行换位思考，看看对方的行为有没有迫不得已的原因。特别是要考虑当自己处在相同情境中时，是否也会表现得比较糟糕。如果回答是肯定的，就不应该过多地指责或责备员工，反而应该更多地体谅和帮助员工。

（二）活动者—观察者效应

与基本归因偏差密切相关的一种归因偏差叫活动者—观察者效应。也就是说，基本归因偏差仅在个体作为观察者解释他人的行为时发生，而在个体作为活动者解释自己的行为时则出现了相反的偏差，即当观察他人的行为时，更多地强调内部因素，而解释自己的行为时，则更多地强调外部因素。例如，别人绩效差是因为他们能力不足或努力不够，而自己绩效差是因为任务太难、运气不佳或领导支持不够。这种现象并不一定是活动者和观察者之间的个人恩怨造成的，可能是因为双方依据的信息不同，也可能是因为双方看问题的

角度不同，但这种偏差常常会导致严重的人际矛盾和冲突，带来个人恩怨，需要在工作中加以注意并力求避免。

（三）自利性偏差

当涉及对自己行为原因的解释时，很容易出现的错误是自利性偏差。个体倾向于对自己的成功做内部归因，而对自己的失败做外部归因。也就是说，成功时，人们接受成功的荣耀；失败时，人们拒绝承担失败的责任。即使是企业的 CEO 们也难免出现这样的偏差。典型的例子是下面给出的绰号“链锯阿尔”的阿尔 • 邓拉普（Al Dunlap）的故事。在组织中，如果领导和下属都受到自利性偏差的影响，那么很容易出现归因矛盾，成功时争抢功劳，失败时互相推卸责任。

“链锯阿尔”的自利性偏差（来源：Nelson & Quick, 1999）

阿尔 · 邓拉普（Al Dunlap）是商业领域中声名狼藉的管理者之一，他的成功和失败一样出名。在他的案例中，成功和失败众所周知，他接受了成功的荣誉，但对失败却谴责他人或环境。这是否是一种自利性偏差？

阿尔 · 邓拉普以缩减规模之王而闻名。20 世纪 90 年代中期，在斯科特纸业公司（Scott Paper），他解雇了几千名员工，削减包括研发、工厂修葺及慈善捐款在内的一切开支，最终令公司业绩改善，股价也上扬了 225%。这些没有人情味、大刀阔斧的措施让他获得了“链锯阿尔（Chainsaw Al）”的称号。将斯科特纸业公司成功改造后，邓拉普又促成了金佰利公司（Kimberly-Clark）对前者的收购。作为 Scott Paper 公司的首席执行官，邓拉普获得 1 亿美元的薪酬。邓拉普写了本名为《商海无情》（Mean Business）的畅销书，这本书的副标题是“我如何拯救运作不良的公司和使运作良好的公司更伟大”，书中大大称赞自己。在书中，他宣称付给他的每分钱都物有所值：“我是我所在领域中的超级巨星，就像篮球领域中的迈克尔 · 乔丹（Michael Jordan），和摇滚音乐中的布鲁斯 · 斯普林斯汀（Bruce Springsteen）”。

陷入困境的阳光公司（Sunbeam）在 1996 年聘用了邓拉普。阳光公司是一家生产搅拌器、电热毯和煤气烤肉架的制造商。邓拉普加盟的消息传出，阳光公司的股票在 6 周内翻了一番。在阳光公司的一年中，邓拉普解雇了该公司 12000 名员工中的一半，削减生产线，展开了又一次大规模的削减措施。一年之后，邓拉普就宣布成功，并且让公司上市，但根本没有买家。相反，邓拉普购买了 3 家消费产品公司。通过玩弄会计花招，收益看起来很好。但是，当他的会计手法被揭穿时，阳光公司欠债达到 20 亿美元，现金流动是负值。这家公司比“链锯阿尔”接手时还要糟糕得多。

对这样的结果，邓拉普指责一名他解雇的经理，指责那些进行“愚蠢的、毛利低的交易” 的下属，甚至指责厄尔尼诺（El Nino）现象——“在暴风雨的时候人们不会想买烤肉架”。当面对董事会时，他宣称一名亿万资产的金融家阴谋低价购买这家公司，因此导演了一系列负面的媒体报道。

公司董事会不再相信他的故事。被指控策划了大额财务骗局的邓拉普最终被阳光公司董事会解雇，而美国证券和期货监管委员会和司法部也将邓拉普列入目标。如果邓拉普坦白承担失败的责任，情况是否会有所不同？我们永远不会知道。

一项对卫生保健公司管理者的研究证实了这种偏差的存在。公司的管理者被要求指出他们的雇员糟糕业绩的原因。管理者宣称，内部原因（雇员缺乏努力或能力）是雇员糟糕业绩的根源。这是基本归因错误的表现。而当要求雇员解释自己的业绩问题时，他们责怪缺乏来自管理者的支持（外部原因），这表现了自利性偏差。

上述三种归因偏差具有一定的文化差异。在强调个人自由的文化氛围下，如西欧、美国、加拿大等国家，基本归因偏差更普遍和明显；而在强调团队合作和相互依赖的集体主义文化背景下，如印度、日本、中国、韩国等国家，基本归因错误不那么明显。自利性偏差也有类似的情况，在强调集体主义的东方文化中比在强调个人主义的西方文化中表现得要弱一些。此外，有研究发现，即使在亚洲国家之间也存在差异。例如，印度流行宿命论的观念，人们相信命运决定了一切发生的事情，在这种文化背景下的人们倾向于强调行为的外部原因。而在中国，人们强调努力工作是通往成功的必经之路，因此当面临成功或失败时，中国人首先考虑的是他们是否足够努力或态度是否端正。一项研究考察了运动员对体育比赛成绩的归因，发现中国的运动员把成功和失败都归为内部原因。即使是当糟糕的成绩是由于外部原因（如坏天气）引起时，中国运动员仍然做内部归因。尽管这一研究结果未必能推广到工作领域，但至少为归因偏差的文化差异提供了佐证。

三、归因偏差在组织中的表现

个体解释周围事件的方式很大程度上影响他们的行为。人们试图理解行为的原因，是为了更好地预测和控制未来的行为。管理者在很多方面都使用归因。在评估业绩和奖励员工时，管理者必须决定行为的原因和责任的归属。一项关于工作场所性骚扰申诉问题的研究发现，当收到这类申诉时，管理者首先需要确定是否存在性骚扰以及谁应该为此负责。研究发现，一些性骚扰事件是由于归因错误造成的。男性很容易把女性的亲密行为归结为性挑逗，这种错误归因造成男性进一步的举动，经常引发女性指控他性骚扰或强奸。当男性处在较高的地位时，这种错误归因更容易发生。一名男老板很可能会误解女下属的顺从或友善表现，并认为女性做出这样的行为是为了吸引他。

在工作场合中，解释自己行为的方式会影响随后的行为动机。例如，假设某员工在公司管理层面前做了一次重要的演讲，自我感觉表现良好，老板也认为非常出色。对于此次成功将做何归因？如果员工相信自己的认真准备和预演导致了成功，那么就会当之无愧地接受赞扬，并对将来的演讲很有信心。但是，如果认为自己只是运气好，碰巧那天老板的心情好，那么该员工将没有动机再现成功。

在招聘面试中，应聘者经常被要求解释他先前成功和失败的原因。研究表明，先前成功和失败的应聘者在对负面结果的归因方式上有差异。先前成功的应聘者防御意识不那么强，对消极事件倾向于做内部归因。而先前失败的应聘者倾向于将负面结果归因于他们不能控制的外部因素，留给面试官的印象是，他没有从失败中学到任何东西，而且面试官会担心当工作中出现问题时，这样的个体也会过多指责别人。

归因理论也可以解释相近的绩效评估结果为何会产生有差别的奖励。如果一名主管将员工的良好业绩归因为内部原因，例如努力或能力，那么他会倾向于给员工较高的奖励；但是如果他将员工的良好业绩归因于外部原因，例如别人的帮助，那么他就不会给予员工高奖励。管理者需要经常反省自己对他人行为作出的归因，以纠正出现的偏差。

第三节　知觉和归因在管理中的应用

组织中的人们经常需要互相判断。管理者评价下属的工作绩效和造成绩效高低的原因，员工也要评价同事在工作中的努力和贡献。申请某一职位的应聘者会受到面试官的评价，进入公司后也会立即受到已有员工的评价，这些应聘者和新来者通常希望给他人留下良好的印象。作为一个整体的组织也希望展示给公众一个良好的形象。组织的这些日常活动都需要知觉和归因的参与。本节从几个方面对知觉和归因的有关规律在管理中的应用问题进行分析。

一、招聘面试和绩效评估

在工作环境中，社会知觉产生重要作用的两个情境是招聘面试和绩效评估。在招聘面试中，面试官的关键任务是形成对应聘者的准确印象。但是，如前文所述，作为知觉者的面试官的特点、知觉对象的特点和知觉情境的特点都会影响知觉的准确性。而且，面试官也会受到知觉偏差的干扰，导致对应聘者形成不准确的印象。通常，外表漂亮的人更容易找到工作，薪水也更高，一个主要原因就是面试官受到晕轮效应的影响，认为外表漂亮的人也具有其他一些积极特点，但这种关联很可能是不存在的。一旦面试官形成对应聘者的最初印象，不管这一印象是否准确，也不管印象是好是坏，都会在很大程度上影响面试官接下来与应聘者的交流方式，并最终影响面试的结果。选择性知觉和自我实现的预言也同样会出现在面试情境中。作为招聘者，必须认识到这些可能偏差的影响，避免做出错误的招聘决定。

绩效评估是在一系列与工作相关的维度上评价员工的过程。通常绩效评估每隔一年或半年进行一次，根据评估结果来决定加薪、晋升和培训需要。理想状态下，这一过程应该是完全理性的，导致对每名员工业绩表现如何和应该如何被对待的公平的、客观的判断。但是，从我们对知觉过程的分析可见，绩效评估过程远不够客观。事实上，人们加工、存储和提取信息的能力有限，造成评估他人时倾向于受到偏差的影响。

研究者已经证实了评估过程中存在的几种偏差。例如，研究发现，人们对他人业绩的评估依赖于业绩与人们预期的一致程度。一项研究要求银行经理指出他们期望新来的出纳员工作做得有多好。4 个月之后，要求这些经理评价出纳员的实际工作表现。结果发现，经理们给予那些符合他们先前预期的出纳员的评价最高，而对高出或低于预期的出纳员的

评价要低一些。这样的结果使人不安，因为它们意味着一些员工的业绩提高可能不被识别和接受，甚至被贬低。当然，人力资源管理决定是根据多种来源的信息，并非仅根据一名领导的判断，这种偏差很可能会被发现。但是，这样的研究结果强调一个重要观点：知觉不仅仅依赖于知觉对象的特点，还依赖于知觉者。

人们还发现了绩效评估中的几种归因偏差。当上级评价下属时，他们之间越相似，上级给出的评价分数越高。这种相似可以是工作价值观、习惯和工作方式上的，也可以是年龄、种族、性别和工作经历上的。下属也更加相信被他们知觉为与自己相似的领导。造成的结果是，相似的上级和下属之间的关系更积极，上级对相似下属的评价更高。一项研究考察了绩效评估情景下的相似效应，调查了不同职业的 111 对上级和下属。结果发现，员工为了使领导对自己形成正面印象所做的努力越多（如帮领导的忙和同意领导的观点），领导就越认为下属与自己相似。而领导越认为下属与自己相似，他们给这些下属的工作评价就越高。员工经常强调良好业绩的内因和糟糕业绩的外因，并尽量与上级的解释保持一致，从而赢得好感。两名业绩同样好的员工可能会因为上级对他们业绩的归因不同而获得不同的绩效评定。通常，糟糕业绩被归因于外部不可控因素的个体获得更高的评价，而糟糕业绩被归因于内部可控因素的个体获得较低的评价。

这些研究证明，组织中的绩效评估确实距离完全没有偏见和理性的理想状况有很大差距。实际的绩效评估过程包含了很多知觉偏差的混合，这些影响必须被识别和理解，以利于最终提高绩效评估过程的准确性。

二、个人的印象管理

大多数人都想给别人留下良好印象。通过各种方式，控制别人如何看待自己，尽最大努力按照别人的期望表现自己，这一过程叫做印象管理。通常个体非常注意自己在他人心目中留下的印象，特别是当他人很重要的时候。对于某些人来说，有意地向外界展示良好印象已经成为生活方式的一部分。这些人不断监控自己的行动，注意他人的反应，调整自己的行为以达到满意效果。这类人自我监控的程度比较高，始终在按照外界的期望和要求规范自己的言行。

潜在雇主对应聘者的印象可能依赖于很多微妙的因素，如穿着和谈吐、介绍个人成绩的方式等。这些印象可能是有意造成的，也可能是无意表现的结果。当涉及到招聘访谈时，应聘者通常会做一些事情来促成良好的印象。一项研究对找工作的大学生和负责招聘的公司代表之间的谈话进行了录音。大学生的陈述被编码归类，分析他们使用的印象管理技术。结果发现了几种常用的策略及其使用的频率，如表 7.2 所示。最常用的策略是自我提升，即声称自己拥有很多正面的特点。应聘者通常把自己描述为不辞辛苦的、人际技能熟练的、目标导向的和具有领导才能的。这项研究还发现，应聘者使用这些印象管理技术取得了很大成功。他们对这些策略的使用越多，面试官在几个重要维度上对他们的看法越正面。这项研究不仅证实了应聘者在招聘访谈中确实使用了印象管理技术，而且证明这些策略常常

是有效的。招聘访谈是一个双向的互动过程，在这一过程中，应聘者努力以最好的方式展现自己，而面试官则需要努力识破这些手法，尽量准确地判断应聘者，但如上述研究所证明的那样，这项任务并不轻松。

表 7.2 应聘者有效展示自己的技术（来源：Stevens & Kristoff, 1995）

印象管理技术	描述（例子）	使用频率
自我提升	以适合当前情境的正面方式描述自己（例如“我工作很努力”）	100%
个人故事	描述使自己显得很优秀的过去事件（例如“在我过去的工作中，我会在需要的时候加班到任何时间”）	96%
观点一致	表达对方可能会持有的观点（例如同意面试官所说的话）	54%
自我标榜	声称成功的过去事件有自己的功劳（例如“销售上升中有 90%是我的作用”）	50%
他人提升	在陈述中奉承、赞美或恭维对方（例如“我对你们公司近来的业绩增长印象深刻”）	46%
夸大	夸大某一积极事件（例如“我们的部门不仅有所改善，而且是整个公司中最好的一个部门”）	42%
解决障碍	描述面临可能降低成绩的障碍时，如何成功地加以解决（例如“尽管我做了两份兼职工作，我还是获得了平均 3.8 的绩点”）	33%
辩解	承认个人的不良业绩，但否定其负面含义（例如“我们的团队获胜的次数不是很多，但重要的是我参与了”）	17%
借口	拒绝为个人的行动负责（例如“我没有完成申请表是因为就业中心的表已经发完了”）	13%

印象管理策略大抵可以分为两类：一类属于自我美化，即传达自身的积极信息，努力增加自己的吸引力；另一类属于他人美化，即通过各种途径努力使对方感觉良好。这两种策略可能反映了不同的目的，前者是希望自己看起来有竞争力，后者则希望让别人喜欢。

自我美化的具体做法包括通过时髦的穿着、整洁的修饰和各种装饰来美化外表，也包括以肯定的方式描述自己，如表 7.2 中的自我提升、个人故事、自我标榜、解决障碍等。一种间接的自我美化手法是提及自己与重要人物的联系，借此来美化自己，抬高自己的形象。

他人美化的重点是引发他人积极的情绪和反应，使对方感觉良好。研究表明，对方积极的反应会使得他们喜欢自己。最常用的他人美化策略是逢迎或他人提升，即称赞他人、称赞他们的特质或业绩或者称赞他人所属的组织。其他的他人美化策略包括：对他人的观点表示赞同、帮助别人、对他人感兴趣以及询问他人的意见并给以反馈。通过频繁的目光接触、同意的点头和微笑也可以表达喜爱，赢得好感。

值得注意的是，一些员工可能会进行消极的印象管理。建立消极印象的方法包括降低业绩、不尽最大努力工作、躲避工作、表现恶劣态度以及宣传自己的弱点等。为什么这些人要让别人觉得自己很糟糕？有时是为了躲避额外的工作或某项特定的任务，也可能是为了被解雇或者吸引别人注意。

印象管理策略是否有效？研究表明，这些努力是有用的，成功进行印象管理的人在很多情况下都表现出了优势。多数研究集中于招聘访谈，结果表明进行印象管理的应聘者更有可能获得职位。而且，进行印象管理的员工在绩效评估中获得的评价更高。一项涉及 1400 名员工的大型研究发现，在各种工作中，包括印象管理在内的社会技能是工作业绩和晋升潜力的唯一最好的预测指标。这就是说，如果策略使用熟练而且适当，那么确实可以提高个人的形象。但是，这些策略的使用也包含潜在危险，如果使用过度或不适当，会导致相反的结果。过多的印象管理会给人留下有意操纵别人或不诚恳的印象。在工作环境中媚上欺下的员工给人的印象非常糟糕。

总之，印象管理策略的使用可能会影响招聘决策或绩效评估的准确性，组织管理者需要认识到这一点，避免过度地受其影响。另外，这些印象管理的技术如果被适当使用，也可以在组织中获得成功或与客户建立良好关系。

三、组织的印象管理

不仅个体需要培养他人对自己的良好印象，组织也需要培养良好的公司形象。组织给人们留下的印象很大程度上影响这些个体与组织的关系。在公司招聘中，不仅应聘者想给未来的雇主留下好印象，雇主也希望他们的工作岗位能被最好的应聘者接受。

一项研究证实了公司形象在这类情境中的重要性。这项研究发现，公司的形象与人们寻求被它雇佣的兴趣密切相关。一家公司的声望越高（根据《财富》杂志的调查），大学高年级学生为它工作的兴趣就越高（2006 年《财富》杂志调查得出的全球最受赞赏的公司排名见表 7.3）。由于公司必须有效地雇佣潜在的员工才可能顺利运作，因此了解到底哪些因素会影响公司形象是很重要的。Gatewood 等人发现，公司形象与人们拥有的关于它的信息的数量（例如，来自先前的工作经历或大学就业指南上的广告）呈正相关。一般来说，较长的广告与更积极的形象相关联，也就是说，不仅广告内容很重要，广告的长度也很重要。由于招聘广告通常会强调被公司雇佣的优点，因此更长的广告列出的优点更多，从而制造了更强烈的正面形象。并且，如果人们相信更长的广告反映了公司赢得优秀员工的承诺（因为公司愿意在广告上多投资），他们更有可能把这家公司作为潜在的工作场所。

组织用于提升公司形象的另一种手法是他们的年度报告，即公司提供给股东的关于公司运作和财政状况的正式报告。根据传统，这些报告是非常精美的小册子，包含精致的照片和眩目的图像以及成功的装饰，用于保持和提升投资者的信心。但是，很多公司省下了这笔费用，代之以不加修饰的年度报告。他们这样做并非是为了节省费用，而是为了在投资者心目中形成节俭的形象。由于当前的投资者寻求的是价值，很多公司都不遗余力地在投资者心目中培育他们不浪费钱财的印象。在精致的年度报告上挥霍金钱来显示成功，可能会让人怀疑利润都去了哪里。因此，不管这些报告精致还是朴实，结论是相同的，年度报告是用来培养“正确的”公司形象的手段。组织如同个人一样，通过给他人以良好的印象可以获益，他们都在努力地做到这一点。

表 7.3　全球最受赞赏的公司（来源：Demos, 2006）

排　　名	公　　司	所 在 行 业	所 在 国 家
1	通用电气（General Electric）	电子	美国
2	丰田汽车（Toyota Motor）	汽车	日本
3	宝洁（Procter & Gamble）	家居和个人用品	美国
4	联邦快递（FedEx）	快递	美国
5	强生（Johnson & Johnson）	制药	美国
6	微软（Microsoft）	计算机	美国
7	戴尔（Dell）	计算机	美国
8	伯克希尔·哈撒韦（Berkshire Hathaway）	财产与意外伤害险	美国
9	苹果电脑（Apple Computer）	计算机	美国
10	沃尔玛（Wal-Mart Stores）	综合商业企业	美国

本章小结：

1．社会知觉是个体收集、整合和揭示关于他人信息的过程，人们的行为是以社会知觉而不是现实为基础的。

2．自我知觉是人们对自己的认识，包括自我概念和自尊。人们对自我的认识不一定准确。尽管自尊高有一些优点，但自尊过高会造成自利性偏差，对组织造成不利影响。

3．知觉者、目标个体和情境的特点都会影响社会知觉。

4．人们对他人的判断存在一些系统偏差，包括第一印象、晕轮效应、选择性知觉、自我实现的预言等。

5．刻板印象是人们对某个特定群体成员持有的固定看法。刻板印象有一定的根据，优点在于节省认知资源。但是，刻板印象常常造成人们忽视个体差异，产生判断偏差。

6．帮助人们克服社会知觉障碍的方法包括识别和面对刻板印象、考虑客观信息、避免匆忙做出判断等。

7．归因是人们解释自己和他人行为原因的过程。对应推断理论认为，人们根据行为推测他人的特点。因果推断理论认为，人们根据区别性、一致性和一贯性信息解释行为原因。

8．常见的归因偏差包括基本归因偏差、活动者—观察者效应和自利性偏差。在解释他人行为时，人们往往低估情境的作用，高估个人特质和态度的影响。对自己的行为进行解释时，人们倾向于对成功做内部归因，对失败做外部归因。

9．在组织招聘面试和绩效评估中很容易受到知觉偏差和归因偏差的影响，必须对这些偏差加以识别和排除。

10．人们希望控制他人对自己的印象。印象管理的策略包括两类：一类是自我美化，即传递自己的积极信息；另一类是他人美化，即让对方感觉愉快，从而喜欢自己。

11．组织需要保持良好的公司形象，才能招聘到杰出的员工和使顾客和投资者满意。

本章思考题：

1．试分析影响人际知觉的三大因素。
2．试述社会知觉偏差及其在组织管理中的表现。
3．举例说明如何纠正组织情景中的知觉偏差。
4．试述归因偏差及其在组织管理中的表现。
5．举例说明如何纠正组织情景中的归因偏差。
6．试述印象管理的概念、种类和一般方法。
7．分析印象管理的优点和缺点。

案例分析：

别去指责沃尔玛（来源：Colvin, 2006）

罗伯特　格林沃德（Robert Greenwald）新拍摄的记录片《沃尔玛: 低价商品的高代价》以沃尔玛为题材，沃尔玛的管理层担心这部影片会像迈克尔 · 摩尔（Michael Moore）有名的反通用汽车公司电影《罗杰和我》一样，把沃尔玛推上批评的风口浪尖。面对这种情况，我给沃尔玛 CEO 李斯阁（Lee Scott）及其团队的第一个建议就是: 不要再为这部电影感到焦虑。摩尔的电影很吸引人，但这部片子完全没有摩尔电影所具备的幽默感、技巧和故事性，令人悲哀。那些本来就厌恶你们的人会喜欢这部电影，不过除了他们，没有人能耐着性子把整部片子看完。我的第二个建议是要好好思考一下影片里表现出来的问题。这是经济全球化给我们这个时代带来的社会大动荡的反映。尽管它可能很糟糕，但这部电影的确反映了那些厌恶并忍受着新旧世界交替折磨的人们的心声。他们为数众多，并希望媒体、市场和政府能够听到他们的声音。作为世界最大的企业，沃尔玛不可避免地成了他们最好的目标。

他们不满的原因很简单。在经济全球化新体系里，调整供应链和分销网络的效率和精确度与以前相比简直是无法想象的，零售业巨头的商品价格也随之降到很低。沃尔玛的批评者（包括这部新电影）花费了大量的笔墨，大肆渲染这家公司是如何冷酷无情地将其他小城市的商场挤垮的。这个论断让人无法苟同，它有一个明显的谬误，那就是沃尔玛无法把任何一个竞争对手从这个行业中赶出去，能这么做的只有消费者。数百万消费者心情愉悦地驾车离开那些小的商场，是因为他们想省些钱。当然，吸引消费者的不仅仅只有沃尔玛。家得宝（Home Depot）和劳氏（Lowe's）对其他小的五金超市，Zales（美国最大的珠宝连锁店）对家庭式小型零售商的珠宝店，Sports Authority（美国最大的体育产品零售商）对老式的体育用品零售商……都是如此。它们都在用突飞猛进的电脑和通讯技术创造着前所未有的商业模式，为消费者带来他们想要的东西，而且这种趋势还会继续发展下去。

这种新体系同样使得雇主们不可能再像以前那样花费人力成本。对于很多美国人来说，新体系最关键的部分或许就是名副其实的全球劳动力市场的到来。在这个市场中，全世界的工人们都在一起竞争。当然，这并不意味着孟买的某个人能直接获得沃尔玛商场大堂收

银员的职位，但劳动力的角色并不是固定的，比如一个人可以在商场里工作，也可以在工厂或是办公室里工作。因此，在全球竞争的影响下，对工厂的工人和那些可以在海外开展业务的相关行业员工而言，他们的工资下降了。这对于沃尔玛的员工和将来可能成为沃尔玛员工的人来说是个坏消息。

纪录片中大部分内容涉及了沃尔玛员工的医疗保险问题。很多沃尔玛员工没有得到良好的医疗保险，或者根本就没有医疗保险。我们再来看看 2005 年的情况。个人医疗保险正变得越来越少。由于经济全球化，不仅美国工人要与中国工人们竞争，沃尔玛也在与全球的竞争对手进行资本上的较量，美国公司不能再为全世界最昂贵的医疗保健费用买单了。沃尔玛并没有责任，要怪就只能怪美国政府在国家医保计划方面的无能，无法卸下压在企业肩上的沉重负担。

电影还提到了一些关于沃尔玛的经理们非法操作的行为。这种行为显然是不可原谅的。关键在于这种行为是否是系统的、有组织的。电影里暗示了这一点，但是没能证实。如果没有更进一步的证据的话，对于这个问题，任何人都不应该再指手画脚了。这对沃尔玛的影响可不像体育比赛里不做不利判决那么简单。(在体育比赛中，当裁判没有充分的证据对可疑情况做出判决时，裁判对该运动员一般不会做出不利判决。)沃尔玛的增长速度已经缓和下来了，投资者向它施加压力，要求改善业绩。当这种压力被层层传递至底层商场的时候，经理们做了一些他们不该做的事情也就不足为奇了。

如果这些事情发生了，而沃尔玛没有加以妥善解决，那么它们将导致可怕的后果。这是一场战争，没有什么东西规定了沃尔玛一定能赢。这种不满的影响将给沃尔玛的竞争者们可乘之机，逐步把沃尔玛挤下宝座，退化成美国几大零售商之一。评论家们的看法并不是重点所在。正如电影结尾所说，这部电影带来的最大危害是，它告诉人们只要阻止沃尔玛，就是赢得了胜利。这是种错觉，不管你对沃尔玛是喜欢还是讨厌，也不管是谁造就了它，能够适应正在到来的经济全球化的新世界，才是真正的胜利。

问题：

1．本文作者赞同格林沃德对沃尔玛的看法吗，为什么？
2．你认为格林沃德还是本文作者对沃尔玛的看法更加准确？
3．沃尔玛的雇员和顾客对沃尔玛的看法会有什么样的差异？
4．沃尔玛应该采取什么样的印象管理策略来建立在人们心中的良好印象？

第八章　组织中的记忆与学习——知识管理

谈起汽车，大家首先想到的是美国通用、德国大众、日本本田等世界知名品牌，而很少想起国内的汽车行业，因为国民对中国的汽车制造业几乎绝望了。就在这前景如此渺茫的中国汽车行业中，还有一丝微弱的曙光在闪亮，那就是中国一汽海马汽车的逆市飞扬：2004 年产销乘用车 6.6 万辆，2005 年增至 8 万辆，位居中国机械工业 100 强，成为中国乃至世界一流的整车制造企业。在日趋白热化的市场竞争中，海马汽车为什么能立于不败之地？先进的生产技术、优良的资产结构以及舒心的工作环境起着非常重要的作用，而最为核心的是海马人建立了统一的知识体系和牢固的知识管理的平台，形成了海马知识管理的文化氛围，提升了企业的核心竞争力。那么，何为知识管理？为什么知识管理的作用如此巨大？怎样进行知识管理？本章将就这些问题进行讨论。

第一节　知识和知识管理

一、知识的含义及类别

今天，人类社会已经逐步演变到了知识经济时代。知识作为一种特殊的生产要素，继土地、劳动力、资本等传统生产要素之后，成为理论和实践关注的焦点。昨天自然资源决定一切，今天知识决定一切。因此，一场研究知识的热潮逐渐展开，形成了看待知识的几种观点。不同的学科有不同的看法，国内与国外也有不同的见解。从学科方面来看，认识论、经济学和信息论的观点是不同的。认识论认为知识就是认识、经验，是意识的反映，是观点的总和；经济学认为知识是产生生产力的资本；信息论则认为知识是浓缩了的系统化的信息。国外学者认为知识时代的知识是一种有组织的经验、价值观、相关信息及洞察力的动态组合，它所构成的框架可以不断地评价和吸收新的经验和信息。它起源于并且作用于有知识的人们的头脑。国内学者认为知识经济时代的知识应该是经过人的思维整理过的信息、数据、形象、意像、价值标准以及社会的其他符号化产物。

随着对知识内涵的认识加深，人们从不同角度对知识进行了分类。

德国哲学家马克斯·舍勒按照知识所涉及的领域不同，将知识分为应用性知识、学术性知识和精神性知识。

经合组织（OECD，Organization of Economic Cooperation and Development）从知识的功能角度，把知识划分为事实知识（Know-what），即关于历史事实、经验总结和统计数据的知识；原理知识（Know-why），即有关自然、社会和人的思维运动的规律和法则的原理

性知识；技能知识（Know-how），即技能和诀窍方面的知识；人力知识（Know-who），即关于谁知道什么和知道怎样做的知识。

从认知加工和社会学角度上看，可以把知识分为显性知识（Explicit Knowledge）和隐性知识（Tacit Knowledge）。所谓显性知识，就是指外在的知识，它可以被编码，是可供人们交流、存储、传播和复制的知识。例如，以专利、科学发明等形式存在的知识。隐性知识或称为隐含经验类知识，主要是指高度个体化，存储于人的大脑，难以编码和形式化的知识，是一种只能意会不能言传的知识。例如，个人的经验、技能、智慧等。显性和隐性知识的划分打破了传统上人们对知识的刻板认识，承认了未经系统化的经验类知识。如果把显性知识比作“冰山的顶端”，隐性知识就是隐藏在水面下的“宝藏”，是社会财富的主要源泉，对它的挖掘和利用能力，将成为个人和组织成功的关键。但如何将知识的巨大潜能发挥出来呢？这就需要知识管理。

二、知识管理的概念和方法

（一）知识管理的概念

惠普公司前 CEO 卢 • 普拉特（Lew Platt）曾说过：“如果惠普集团知晓惠普善于什么，我们的盈利水平就会增加三倍。”这就提示企业不仅要拥有知识，而且要知道如何运用和管理好知识。对知识管理（Knowledge Management，简称 KM）问题的探讨可以追溯到 20 世纪 30 年代，哈耶克（Hayek）在《个人主义与经济秩序》一书中论述了“知识要在社会中利用”的新观念，这也是知识管理思想的萌芽。随后，经过无数研究者们的努力，知识管理的概念在 20 世纪 90 年代初成型。综合起来主要有以下几种观点：

（1）马奎特（Marquardt，1996）认为，知识管理就是知识获取、知识创造、知识传播与应用、知识存储等活动过程。

（2）威格（Wiig，1998）指出，知识管理主要包括四个方面：自上而下的检测，推动与知识有关的活动；创造和维护知识的基础设施；组织和转换知识；提升知识的使用价值。

（3）法拉普多（Frappuolo）则将知识分为外部化、内部化、中介化和认知化四种功能。其中外部化是指从外部获取知识，分别将其组织起来；内部化和中介化所关注的分别是显性和隐性知识的转移；认知化则是将上述三种功能获得的知识加以运用，是知识管理的最终目标。

（4）盖特纳（Gertner，2000）把知识管理定义为：通过对企业组织能力的提升，成功地达到对企业信息的掌握、鉴别、检索、分享与评价。

仔细分析上述学者的观点，可以看出，知识管理不再是传统上对人的管理，而是更倾向于知识的分享、信息的传递、组织的学习、智力资本以及绩效的管理，其最终目的就是提升组织的绩效和核心竞争力。因此，本章将知识管理定义为：组织借助于一定的媒介或其他信息传递系统，对知识进行有效的识别、获取、储存、传播、开发和应用，以提高个人或组织创新能力，从而提升组织绩效和核心竞争力的过程。

（二）知识管理的方法

托马斯·亨利·赫胥黎（Thomas Henry Huxley）曾说："人生的终极目的不是获取知识，而是付诸实施。"研究知识管理，最为重要的就是掌握知识管理的有效方法，以运用到实际管理活动中去。国外对知识管理方法的探讨主要来自于美国生产力和质量研究中心。该中心提出了实施知识管理的六种有效策略：构建知识管理的组织体系；加大对知识管理的资金投入；创造有利于知识管理的组织文化；制定鼓励知识创造和转移的激励举措；开发支撑知识管理的信息平台；建立知识管理评估系统。卡尔·弗莱堡罗（2003）也提出了知识管理的十大步骤：界定研究群体；确定实施战略和关键的成功因素；实施知识审计；计算 ROI，即投资回报；制定正确的知识管理计划；识别并认同组织的核心能力；进行知识资源盘存；确定非正规知识实践质量；建立有利于激励方案的生态环境；提供基础结构和优化手段。

国内学者也对知识管理的方法进行了研究。徐勇等（1999）认为，有效的知识管理方法为：设立知识主管；创建动态团队；建立知识创新的激励机制；建设企业内部网络，形成知识交流网；建立递增收益网络；构建动态联盟。王如富等（1999）也提出了知识管理的六种方法：知识编码、组织学习、建构知识库、设立知识主管、应用信息技术、进行基准管理和最佳实践。

尽管许多学者对知识管理的方法和手段进行了比较深入的探索，但是到目前为止还未形成一套完整的、成熟的、切实可行的方法。综合以往研究者的观点，可以看出，要有效地进行知识管理，以下几方面是不可或缺的：（1）创设有利于知识管理的文化。良好的组织文化建设是进行知识管理的前提，要从尊重员工、尊重员工的需要及创设学习型组织入手，提升组织成员内隐知识管理的动力。（2）建设学习型组织。组织学习是实现知识累积、转换和创造，化组织知识为企业最大价值的关键，也是掌握知识管理方法的重要途径。（3）知识编码与构建知识库。知识编码有利于组织更好地学习知识，特别把知识分类储存在可以直接提取的地方——知识库中，这是知识管理的关键所在。（4）建立知识管理的评估体系。知识管理的方法是否有效，必须建立一套合理的评估系统，以便不断地修正模型，更好地进行知识管理。

近年来，虚拟企业知识管理（virtual enterprise knowledge management）成为研究的热点，为合理、有效地进行知识管理提供了新的视角。虚拟企业知识管理中的知识可分为企业内部知识和企业间的网络知识两种。其中，虚拟企业间的网络知识是不同成员企业知识系统中的显性知识和隐性知识有效互动和企业间知识整合的结果。对此种知识的具体管理流程如下：

（1）计划阶段。根据市场需求制定组织目标，在此基础上确定知识管理目标。

（2）实施阶段。运用知识管理工具和手段进行知识管理活动。

（3）调整阶段。对知识资产进行评估，并不断反馈，以调整、修正知识管理目标与计划。

其模型框架如图 8.1 所示。

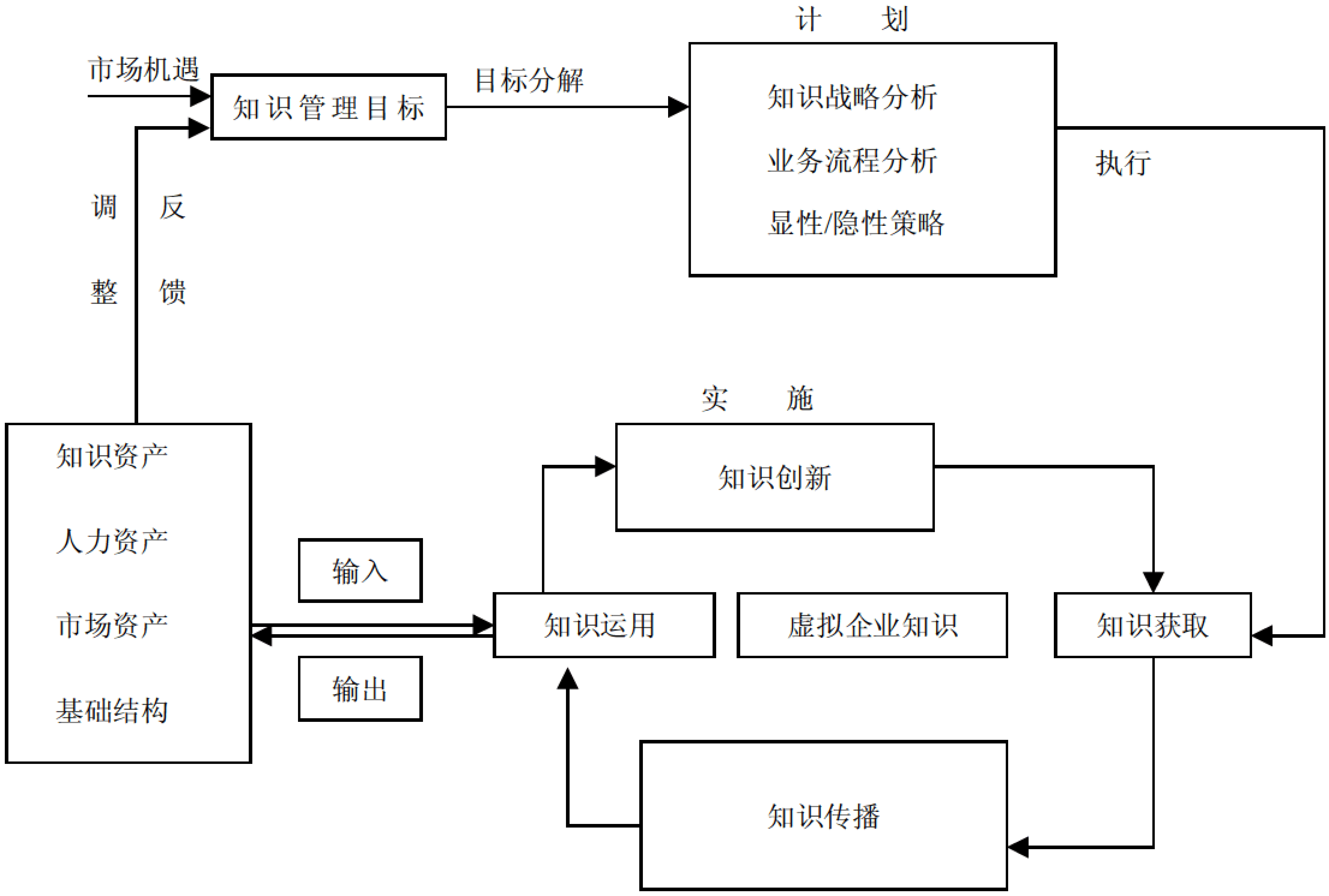


图 8.1　虚拟企业知识管理模型

三、知识管理的相关理论

（一）基于知识链的知识管理理论

该理论以麦克尔·波特为代表，借鉴价值链的概念，提出了知识的价值链。它立足于知识在企业业务中的流动过程，从业务流程的角度把知识在组织内部的流动情况划分为紧密相连的各项活动。“K9 知识链”理论将知识流动的过程分为如图 8.2 所示的九个环节。知识管理就是对不同环节及其相互间的关系进行管理。

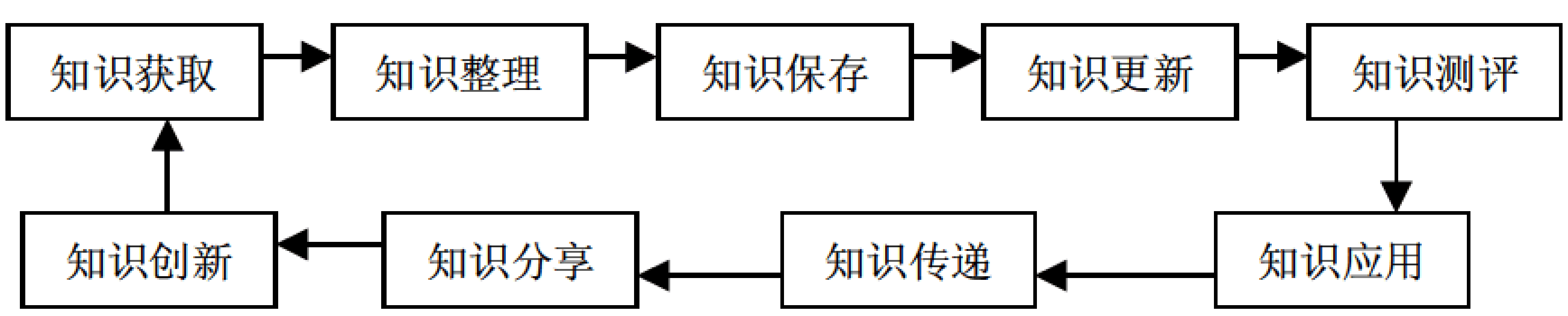


图 8.2　K9 知识链

知识链理论在知识管理流程中加入了一个个的“节点”，便于管理实践中进行控制与评价。其缺点是过分侧重显性知识的管理和创新，忽视了隐性知识的作用，未能探究知识

创新的源泉，在指导知识创新活动中的作用有限。这就呼唤新的模型的产生，因此基于显性与隐性知识流动的知识管理模型应运而生。

（二）基于显性与隐性知识流动的知识管理论

以日本学者野中郁次郎和竹内为代表的研究者，从知识的性质出发，认为知识管理及知识创新的关键就是充分挖掘组织的隐性知识，实现隐性与显性知识的不断相互转化。而这转化并不是单一层面上的单一循环过程，而是呈螺旋式上升的模式。在这种模式中，隐性与显性知识不断转化，从而使组织的知识量越来越大，知识也就从个人水平逐渐上升到部门、团队、组织水平，甚至跨越组织上升到组织之间，达到全社会水平。他们构建的知识循环模式如图 8.3 所示。

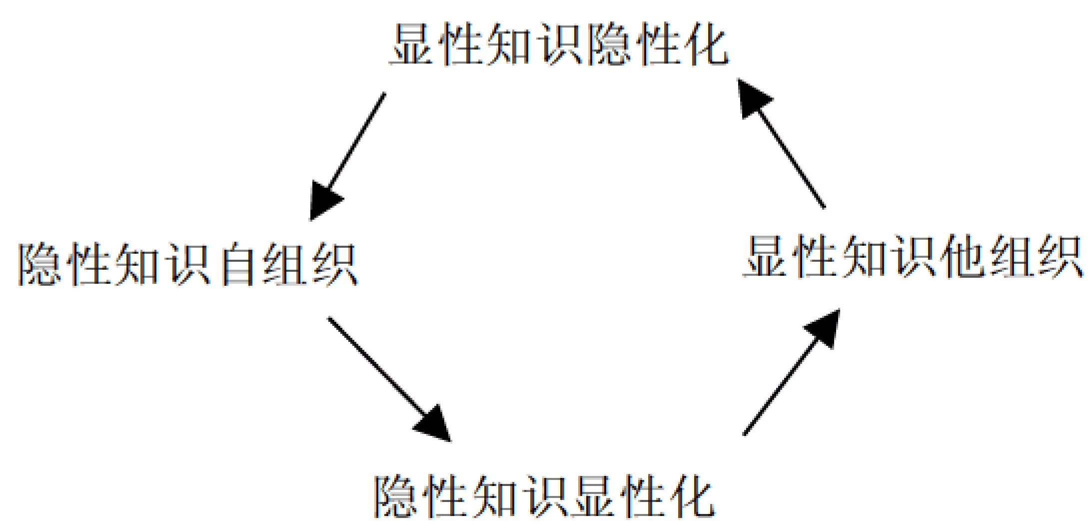


图 8.3　知识循环模式

知识循环模式图中，隐性知识是通过自组织的形式来组织的，表现为动态、模糊与变化。显性知识的组织是他组织，表现为静态、精确与控制。知识的他组织是对现存知识的存储和整合，而知识的自组织是知识意义的倍增和创新。

总之，该模型从知识的性质出发，揭示了组织知识创新的源泉。它认为组织本身并不能产生知识，知识的创新只能从蕴含在员工头脑中的隐性知识开始。因此，企业进行知识管理以及创新管理的关键在于构建一套适应知识积累和创新的制度和环境文化，促进员工头脑中的隐性知识流动和传播，实现隐性知识的显性化。

（三）基于有效处理过程的知识管理论

知识管理不仅仅是一个新的管理问题，而且是一个复杂的社会技术系统。该系统以人和信息为基础实现整合组织知识的过程，主要包括知识表征、知识传递、知识利用和知识发现等对知识的处理过程，如图 8.4 所示。

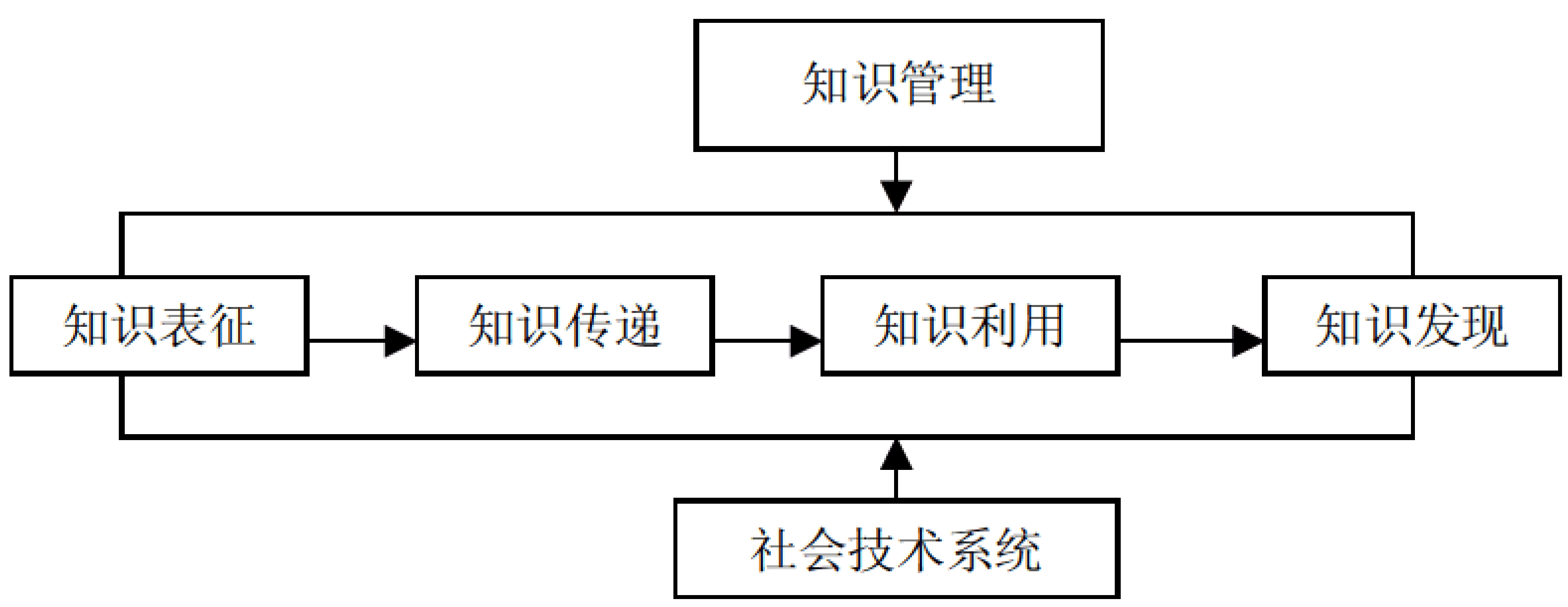


图 8.4　知识管理的处理过程模型

知识表征即知识的储存方式，除了运用传统的方式储存一般的信息和数据外,大量的逻辑、智力以及社会、文化方面的知识要用知识库的方式来储存。如何保存知识是知识管理系统的核心问题。知识传递的手段和工具，是知识管理系统的另一个重要问题。除了传统的方法外，现代信息和网络技术提升了知识传递的效率。知识利用是知识管理的目的。除了研究如何利用知识促进个体发展外，更重要的是研究如何利用知识为组织提供有效服务，如建立决策支持系统等。知识发现是知识创新的主要手段，如何才能在使用知识过程中不断发现对组织有用的新知识，也是知识管理的重要任务之一。

第二节　组织学习与遗忘

联想创办于 1984 年，从十几个人、十多万元创业，发展到今天已有 2 万多名员工，年收入 400 多亿元的大型企业。特别是在收购 IBM 的 PC 业务之后，联想已成为世界第三大 PC 机生产商。联想成功的秘诀是什么？董事长杨元庆告诉我们：“联想要成为 IT 巨头，必须首先是一个非常有心的‘学习者’。”联想的成功是联想人重视组织学习的结果。那么什么是组织学习？为什么它的作用如此巨大？

一、组织学习的概念

学习是日常生活中最常用的概念之一，在心理学上它是指通过经验获得而导致行为模式变化的过程。它本是用来描述个体行为的，这里用来描述组织的行为，实际上是一种类比或借用。因此，除了针对的是组织而非个体之外，组织学习与个体学习有类似的含义。

当将学习作为组织的一种行为时，由于研究者看问题的角度不同，形成了关于组织学习的不同理解。现从组织不断变革以适应环境变化的角度来介绍几种具有代表性的组织学习定义。

阿吉里斯等（Argyris & Schon,1978）认为：组织学习是发现错误，并通过重构组织正在使用的理论而加以改正的过程。

福尔等（Fiol & Lyles,1985）指出：组织学习是通过理解和获得丰富的知识来提高行为能力的过程。

辛格（Senge,1994）把组织学习看成是管理者为了取得组织成员理解，提升管理组织和环境的能力，不断改进决策以提高组织效率的过程。

迈耶斯（Meyers,1990）和应瑛、吴晓波等（2001）认为：组织学习是企业应该拥有的一种能力，即通过观察、评估，并以目的明确的方式对组织内部与外部刺激采取行动的能力。

列维特等（Levitt & March,1988）将组织学习看作是对过去行为进行反思、推论，形成指导组织行为的规范，从而实现组织目标的过程。

尼维斯等（Nevis,iBella & Gould,1995）把组织学习看成是一个系统层面的问题，即组

织基于经验维持或改善绩效的过程或能力。

综上所述，尽管研究者对组织学习的概念存在一定的分歧，但在以下几点上也存在一些共识：组织学习是一个过程，是组织信息获取与传播、知识记忆与运用的过程；组织学习是一种能力，即知识获取、传播与创新和应对内外刺激的能力；组织学习是为达到组织目标而采取的行动，关注的是学习的效果；组织学习还是一个系统层面问题，涉及组织如何学习以及怎样学习的问题。总之，要全面把握组织学习的概念，应该考虑以下几个方面：（1）组织学习的主体是整个组织，不是单个的人；（2）组织学习的客体是知识与信息及其衍生物（如文化、价值、观念等）；（3）组织学习的终极目标是提高组织效率，提升企业的素质。因此，组织学习是组织以系统思考和组织文化的共同愿景为工具，以组织结构为依托，以技术为保障，在持续进行的内外部信息交流中，重新设计以适应不断变化环境的过程。其中，最为主要的是对知识的获取、整理、应用与创新，以指导组织行为，提升组织的核心竞争力与创新力，如图 8.5 所示。

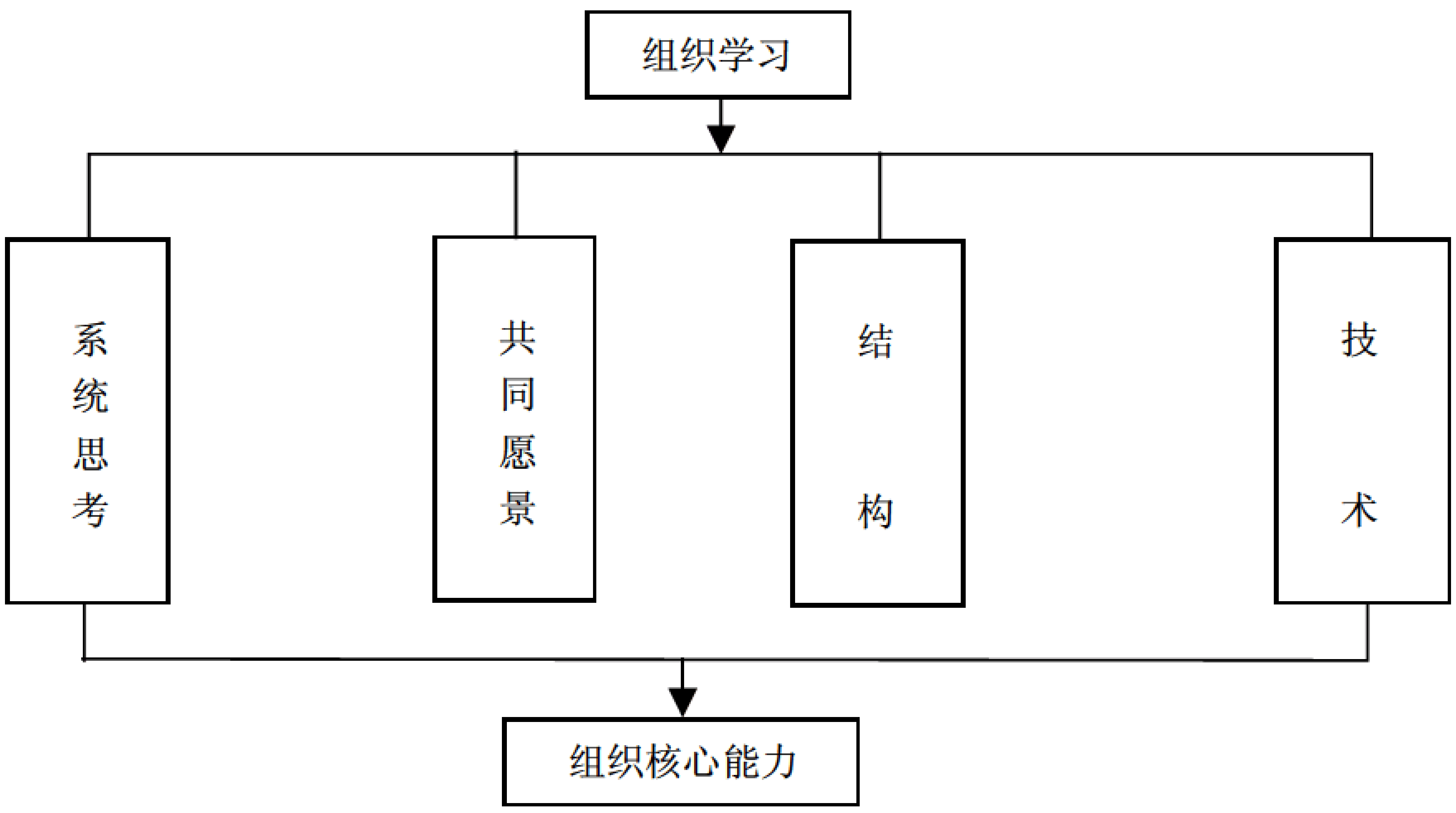


图 8.5　组织学习的概念

二、组织学习的过程模型

组织学习的过程模型试图抽象地描述组织学习的过程，阐释组织学习的构成要素及内在联系，目的在于回答组织究竟是如何学习的这个问题。不同的专家学者从不同的视角出发，对组织学习模型进行了深入探讨与研究，产生了许多理论与应用价值很强的组织学习模型。

（一）经典组织学习过程模型

这是最早、也是最有代表性的组织学习模型。它是由阿吉里斯和谢恩（Argyris & Schon）于 1978 年提出的，把组织学习过程分为如图 8.6 所示的四个阶段。

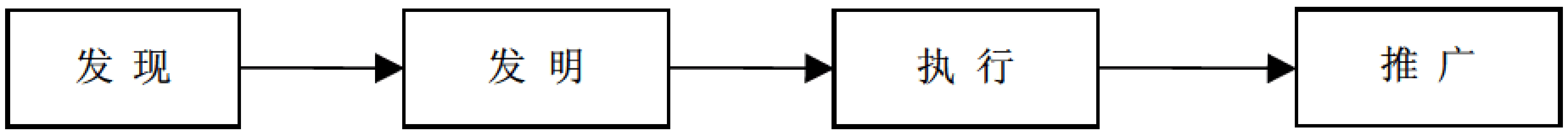


图 8.6　经典组织学习模型

“发现”即找到组织存在的潜在问题或环境中存在的机遇。在“发明”阶段，公司着手寻找解决问题的方法。其解决方法在“执行”阶段得到有效实施，即转化为新的或修改了的操作程序、组织机制或报酬系统。新程序的实施能否保证学习发生在组织水平上？答案是否定的。因此，学习必须从个人水平上升到组织水平，贯穿到组织各部门或组织边界，这就是“推广”。

这一模型的前两个阶段属于适应型学习，不能引起组织原有规范的改变；而后两个阶段组织则产生了新的规范或程序，属于变革型学习。变革型学习需要根据外界环境的变化来调整企业的目标，使企业目标更好地反映外界的需要，是更高层次的学习。

该模型打破了传统适应型学习模型的传统框框，创造性地提出了变革型学习的概念，无疑将组织学习的研究向前推进了一大步。它昭示人们在环境快速变化的时代，组织只有将适应型学习上升为变革型学习，才能应付激烈的市场竞争。

但是，经典组织学习模型也存在一定的缺陷。其一，没有反应组织学习的全过程——缺少学习的反馈环节。没有反馈的学习是无实际意义的学习，也不能反应企业内的真实情况；其二，该模型不能反映组织学习的螺旋式上升的过程，即学习不仅是有反馈的螺旋过程，还是积累组织知识的过程。假如没有以前的知识积累，组织学习根本无法进行。所以，经典的、直线式的组织学习过程模型难以描述组织学习的真正过程和本质。

（二）改进的组织学习过程模型

当经典的组织学习过程模型不能真正描述学习的本质时，改进的组织学习过程模型便应运而生了。该模型在经典的组织学习过程模型基础上，加入了反馈过程，使组织学习过程成为一个闭环，增加了组织学习的有效性。而且，在组织学习的各个阶段都可能发生知识的积累，图 8.7 所示的组织知识库就反映了这种知识的积累和组织学习的螺旋上升过程。

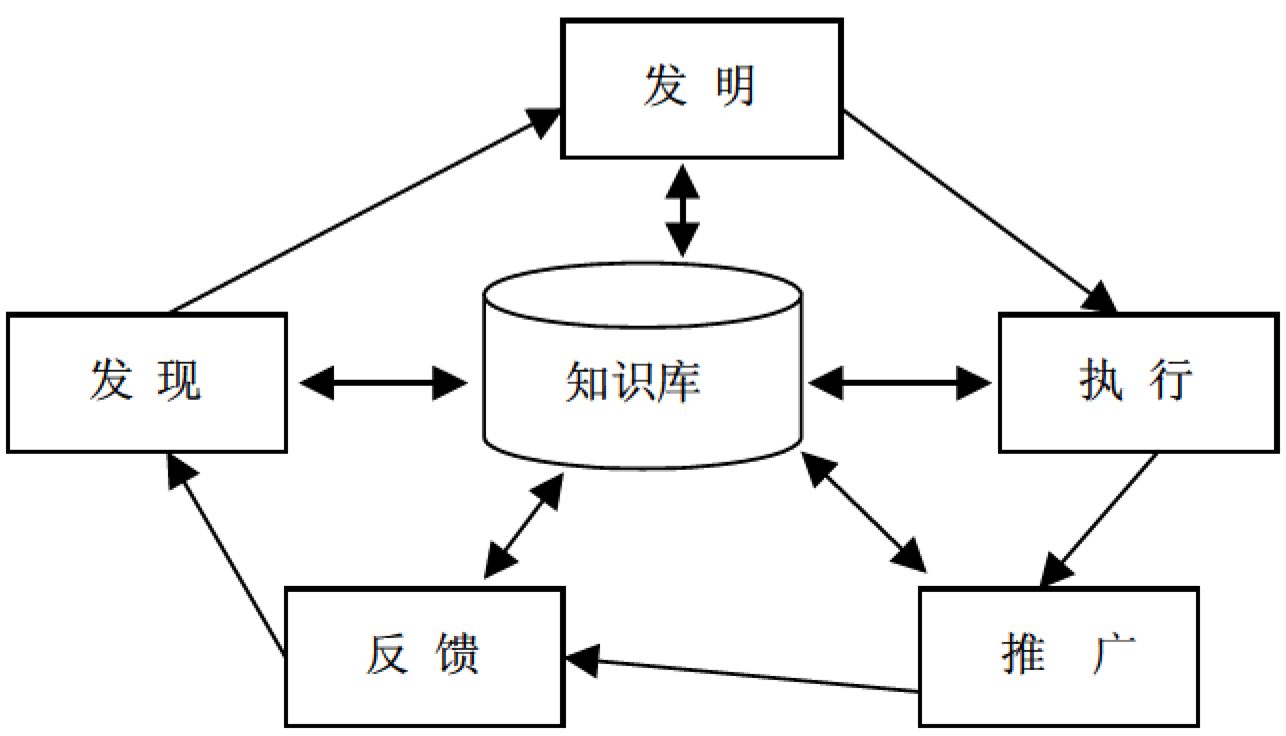


图 8.7　改进的组织学习过程模型

图 8.7 所示的知识库与组织学习各阶段之间的双向箭头，表明每一阶段既可以产生新知识存入知识库，也可以运用知识库中的知识辅助该阶段进行判断与学习。特别要注意的是，组织知识库不一定是一个在技术上实际存在的知识库或数据库，它会以各种不同的形式存在于组织中。知识的一部分可以保存在组织每个成员的头脑中，也有一部分可以存在于组织的经验、数据库、操作规范或文化中。组织在学习的过程中积累了这些知识，即把各阶段发现的知识存储在了组织的一个“知识库”中，以备随后需要时使用。上述组织学习模型反映了组织学习的五个阶段，每个阶段产生的知识存入“知识库”，而“知识库”中的知识也对每个阶段的学习行为产生影响。

改进后的组织学习模型，更真实地反映了组织进行学习的过程。这样可以更加清晰地考虑组织在学习中发生的问题以及解决的方案。企业也只有认识了组织学习的全过程，才能更有效地进行学习，制定推进组织学习的措施。

（三）“SECI”螺旋上升式组织学习模型

这是日本学者 Nonaka（1991）等首先提出的。他们认为组织内的知识可以分为隐性知识和显性知识。隐性知识是存在于组织个体头脑中的、个体的、有特殊背景的知识，即组织中每个人所拥有的特殊知识。它依赖于个人的不同体验、直觉和洞察力。而显性知识是指在个体之间能更系统地传达、更加明确和规范化的知识。在此思想的基础上，Nonaka 构建了“SECI”螺旋上升式组织学习模型，也称为知识转化的组织学习模型如图 8.8 所示。其中“S”指社会化（Socialization），即组织学习是从个人间共享隐性知识开始的；E 则是指外在化（Externalization），也就是隐性知识在团队内共享后经整理被转化为显性知识的过程；C 称为合并（Combination），是指团队成员共同将各种显性知识系统地整理为新的知识或概念；I 即内在化（Internalization），是指组织内的各成员通过学习组织的新知识和新概念，将其转化为自身的隐性知识，完成知识在组织内的扩散。

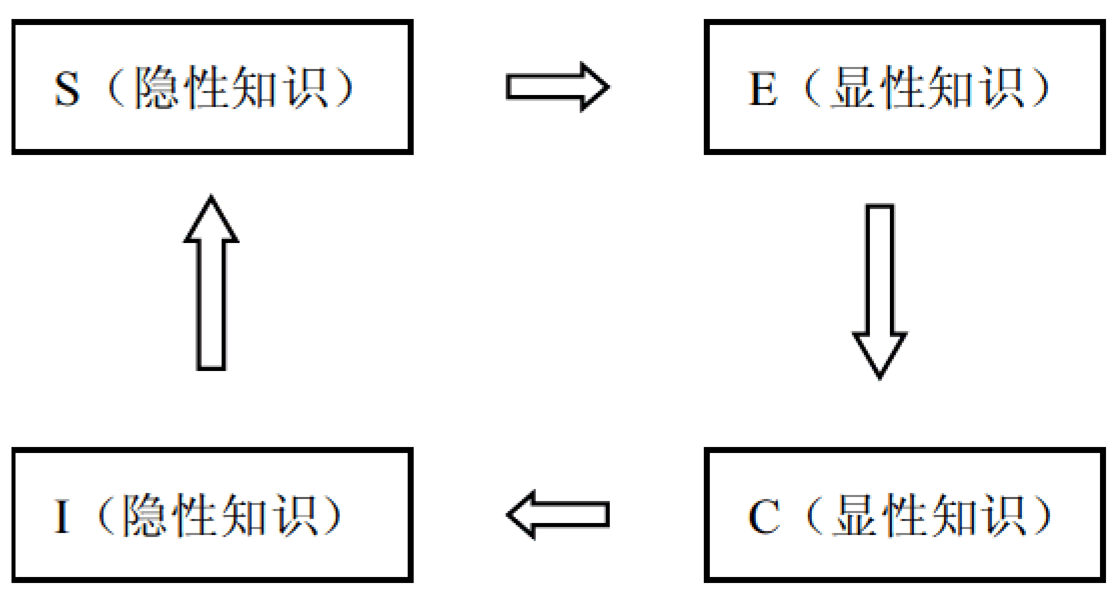


图 8.8　知识转化的四种模式

“SECI”螺旋上升式组织学习模型很好地解决了组织内部知识转换的问题，突破了仅仅把组织学习看作是信息处理过程的局限，而把创新与知识产生建立在隐性知识和显性知识互为转化的过程上，是对以往组织学习模型的创新和突破。但正如日本的组织传统一样，它忽视了组织自身作为知识创新主体的作用，如组织可以设置专门的知识创新部门（如研发部门或知识管理部门）来促进知识的创新。而且，仅仅将组织学习局限于组织内部，缺

乏与外界的知识交换。

（四）组织学习循环过程模型

该模型是迪克松（Dixon，1994）提出的，后来很多学者进行了修正与补充，形成了如图 8.9 所示的组织学习循环过程模型。

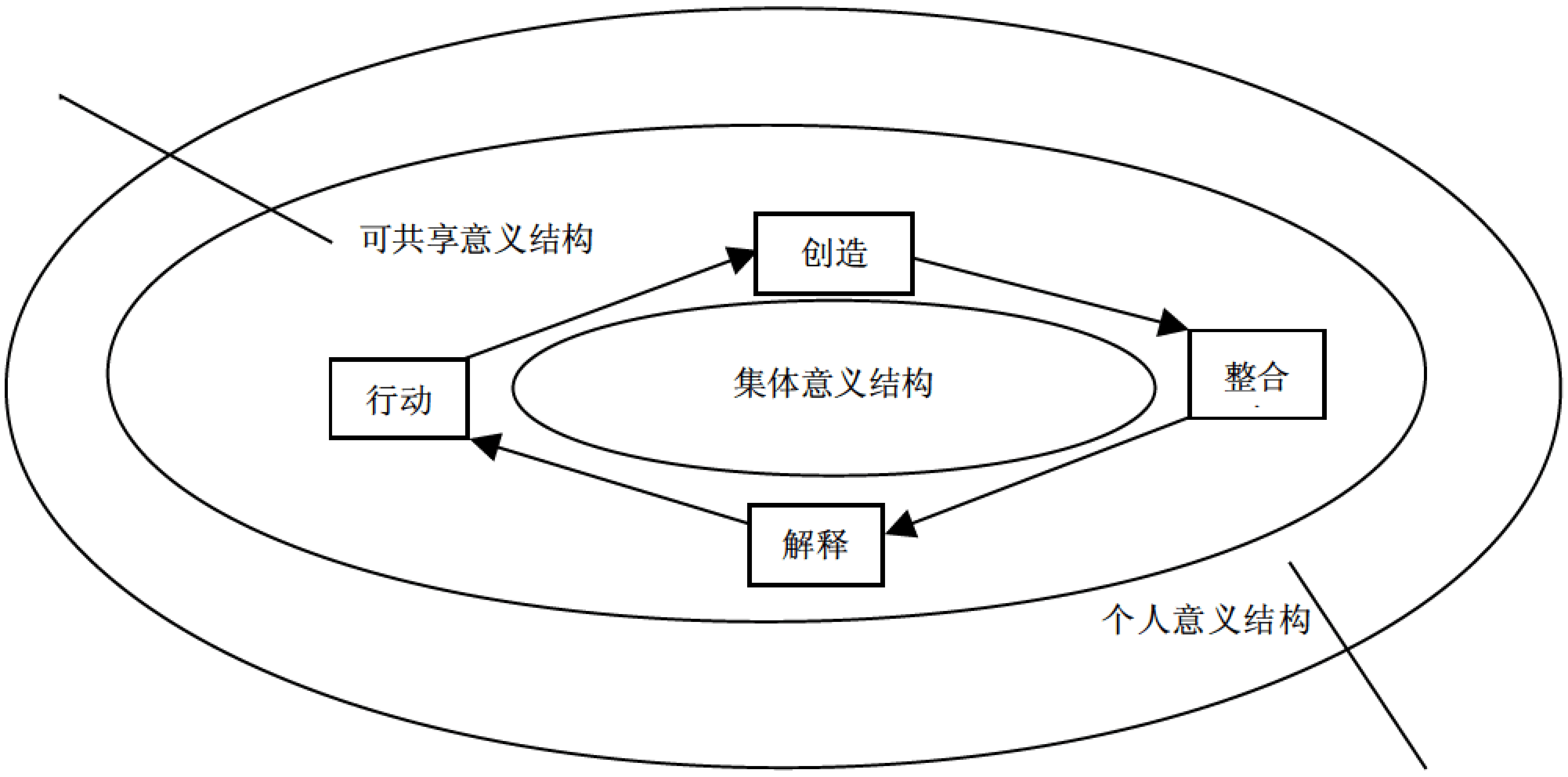


图 8.9 组织学习循环过程模型

该模型把组织学习分为四个循环的过程。学习的第一阶段为创造（generate），即对内外部信息的收集与归纳；接下来为整合（integrate）阶段，主要是运用培训和会议系统的方式将这些信息整合到组织中去；第三阶段为解释（interpret），通过改进的交互作用以及简化的层级体制来解释这一信息；最后一阶段为行动（act），即个人和小组在共同理解的基础上采取适当的行动。

此模型最为重要的贡献在于"意义结构"概念的提出。它取代了传统意义上的"知识"和"信息"，创造性地将意义结构分为个人意义结构、可共享意义结构和集体意义结构三个大部分。个人意义结构与组织中共享部分形成可共享意义结构，而集体意义结构在可共享意义结构的基础上被整个组织吸收，成为组织内大多数成员所认同的价值观和信念。组织学习循环过程模型的目的是使个人意义结构成为可共享意义结构，最终形成集体意义结构。它把个体学习与组织学习有机地结合起来，并强调了组织学习与组织绩效的重要关系，具有一定的可操作性。

三、组织学习中的智障

1983 年英荷壳牌石油公司的一项调查发现，1970 年名列美国《财富》500 强的大企业，有三分之一已经销声匿迹了。企业从诞生到衰亡的平均寿命为 40～50 年。这是为什么？

彼得·圣吉（Peter M.Senge）也曾经追问过："为什么组织中每个成员的智商都在 120

以上，而整体智商只有 62？”。

这一切都是组织学习中的智障——组织中削弱学习、阻碍学习的思维方式、习惯、看法、氛围等在作怪。它妨碍了组织成长，使组织被一种巨大无形的力量侵蚀着，甚至吞没。这就导致组织在传播、共享、储存和应用知识的过程中往往会发生“过程损失”，致使组织学习的效果远远低于个人学习效果的总合。学习智障是企业系统自身客观存在的一种负面因素，仔细分析组织学习智障，了解组织学习智障的类型，有助于挖掘造成学习智障的根源，提高组织学习能力，使组织学习形成良性循环，最终提升组织学习力。

国内外对组织学习中智障问题的研究目前还处于初始和探索阶段，综合所有研究发现主要有以下几种智障负面影响最大。

（一）结构智障

它是指由于组织结构弊端而引起的组织学习智障。组织结构反映的是组织成员的分工与协作关系，其设计的目的是为了更好地、更有效地将组织成员组织起来，为实现组织目标而共同努力。

但是等级森严的组织结构、高度集权的决策方式、一成不变的规则和程序，这些工业时代的“宠儿”，已经不再符合知识经济时代的需要。它们阻碍了知识的顺畅传播、信息的有效反馈、决策的快速传达，使企业反应滞后，从而削弱了其竞争优势。例如，我国国有企业的管理者与员工是典型的上下级关系，而不是同一团队的合作者。员工的报酬不仅取决于个人能力的大小，而且要看其级别的高低，这样就大大降低了员工学习的积极性，严重影响了企业的工作效率。

因此，科学、合理地设计组织结构，简化组织结构的层次以及等级，让信息和知识的传播更加通畅，提升企业反应速度，是消除组织学习结构智障的重要举措；加强员工之间积极而有效的沟通，增强员工的团队合作精神，提高员工学习的积极主动性，形成组织学习的合力，是清除组织学习结构智障的又一重要措施。

（二）文化智障

这里的文化，主要指企业组织文化。它具有很强的影响力和渗透力，在组织学习模型中处于基础性地位。有人把组织文化比喻为一张弓，如果我们拉得太紧，它会试图弹回来，这样我们反而被击中了。

组织学习的性质和方式在很大程度上是由组织文化决定的。但是，大多数企业文化既不排斥学习，也不鼓励学习，处于追求平稳，害怕冒险、创新，拒绝信息共享，免予承担责任的状态。例如，组织中的成员安于政治化的环境，被政治虚伪、隐瞒与势力所熏染，导致在组织中如入鲍鱼之肆，久而不闻其臭。这样就会阻碍新思想和行为的推行。

组织文化的智障首先表现为对企业员工缺乏约束力。组织文化最大的魅力就是对企业员工的内隐约束力，它就像万能胶一样让员工牢牢地粘在企业周围。但是一旦缺失对员工的无形约束力，不仅会影响组织成员学习，而且直接关系到组织形象。例如，曾经一度成为中国海外投资楷模的中航油公司，因员工缺乏诚信，违规暗箱操作，被斩落马下，损害了国有企业的形象。

其次，文化智障表现为企业凝聚力差。凝聚力的高低直接关系到组织学习的效果。日本、韩国企业最为突出的特点，就是具有强烈的团队精神，这也是他们在竞争中立于不败之地的重要因素。

再次，不完善的激励机制是组织文化智障的重要表现。薪酬体系的不完善，绩效考核的不合理，晋升体制的欠缺等都是阻碍员工积极性的重要因素，难以形成一种积极向上的组织文化，从而降低了员工以及组织学习的效率与效果。

（三）管理障碍

通常情况下，进行有效的组织学习或建设学习型组织，需要获得高层管理层的承诺。组织学习过程中，领导者是引领组织学习的方向盘。作为设计师，他要通过战略策划工作来对现有的资源进行整合，设计出组织的结构与组织的政策、策略，提出组织发展的基本理念、设计组织学习的过程。但是管理层往往不能根据组织的实际情况来制订管理方案，致使部门之间不能很好地协调，责任不明确。具体表现为管理的观念落后，责权利不明确，领导的授权不足等。特别是当领导者信息过载时，就会影响组织决策，进而影响组织学习效果。此外，以下几种情况也可能造成管理智障：

（1）从众效应。组织决策如果经常是在不容商量的气氛中做出的，组织成员就会表现出不情愿的从众行为，所以即使对错误的决策，也表示一致同意。强迫达成的共识会抑制成员的积极思维和有创意的想法，引起群体思维倦怠，影响组织学习的过程，丧失组织学习的机会。

（2）习惯性防御。个体免于说出真正的想法，用来保护自己或他人，是一种根深蒂固的习性。它在个体最有价值、最核心的思想和观念周围形成一层保护壳，让这些思想和观念无法表露和通达他人。例如，组织学习过程中经常遇到的障碍是习惯性防御让大家都保持缄默，知识和信息无法交流。

（3）共同悲剧（tragedy of common）。它是生态学家哈定（Garrett Hardin）提出来的一个概念，指行动对每一部分都是有利的，但对整体却是有害的。在一个组织中，个体常常会脱离组织，在短期内通过自利行动而获利，获利会引发更多的活动，进而获利更多。但是当他们发现出现共同错误时，要拯救整体已经太迟了，紧接着是所有的个体都跟着灭亡。因此，组织学习过程中，成员要为整体利益而不断向决策者提供自己的想法，以免自己虽然获得了微小的收益，但组织却遭遇巨大的灾难。

（四）心智智障

心智智障是阻碍组织进行有效学习的重要障碍，具体体现在以下几方面：

（1）局部思考。它是指不从全局、整体和事物的普遍联系考虑问题，而是片面地、局部地、孤立地考虑问题。我们常只专注于自己的决定，而忽略自己的决定对他人有何影响。当组织中的人只专注自身职务时，他们便不会对所有职务互动所产生的结果有责任感。现代组织多是按功能导向进行设计，分工比较细，更加深了这种学习智障。因此，在组织学习过程中，我们必须学会系统思考，放弃必定要由某个人或某单位负起责任的假设，共同分担系统所产生的问题。

（2）专注于个别事件。即组织成员关注的是问题或事件的本身，而忽略事件或问题是由缓慢、渐进的过程形成的。这一专注个别事件的习惯常常使缓缓而来的致命威胁习而不察，佛罗里达风暴的“蝴蝶效应”正是强调个别事件的结果。这就要求我们在组织学习的过程中，适当放慢认知变化的步调，注意细微的、不寻常的变化，察觉构成最大威胁的渐进过程，寻找出隐藏于事件或问题背后的真正原因，否则无法避免“温水煮青蛙”的命运。

（3）利己主义归因。这是人们在进行原因归纳时，出现的一种偏差。即总是将成功的结果归结于自身，而把失败的结果归因为他人或外部环境。组织学习过程中出现的新情况、新问题阻碍学习的进程，组织成员往往会从引发障碍的客观因素找原因，而不愿自身反思。这往往导致组织不能寻找到阻碍组织学习的根本原因，从而降低组织学习的效率与效果，影响企业的工作效率。

（4）消极思维。遇到组织学习困难时，不能从整体上去思考，而是故意拖延，往往使问题扩大，最终导致危机。所以，要实现组织学习的有效性，必须让组织的思维积极起来，具有前瞻性看问题的能力，掌握整体思维的方法，并积极行动起来，将危机扼杀在萌芽中。

（5）经验思考的错觉。遇到问题，组织往往会思考以前解决问题的途径，运用以往的经验来决定解决问题的方法。但是，这里有一个前提，就是问题没有超越时空的有效范围，具有完全的相似性。一旦超越这一范围，试图运用经验发现问题、解决问题的方法，就带有盲目性，有时会得到相反的结果。

（五）执行智障

它是阻碍组织学习活动顺利开展的特殊障碍。具体表现为：组织领导力差、组织学习的战略流程不到位、人力资源配置不合理、学习的运营渠道不通畅等。

组织学习过程中，领导的素质以及执行力直接关系到组织学习活动能否顺利进行。在一个领导重视且执行力强的团队中，组织成员就会有积极学习的强大动力。相反，则会使组织学习缓慢而滞后。

企业组织学习的战略流程是关系组织学习活动成败的重要因素。企业学习战略是企业发展的“生命线”，它指导组织学习活动，决定组织学习的方向和企业持续竞争优势的获取能力。制定并切实实施合理的组织学习战略流程，可以将组织学习的主体与组织学习的运营过程有机结合起来，让企业组织学习活动与外部环境紧密衔接，使组织在竞争中持续处于优势地位。

人是组织学习的主体，也是组织学习最关键的因素。组织中人力资源的选拔、培养与配置，是组织学习活动顺利开展的重要因素。人员配置合理与恰当，形成了学习型团队，就会让组织主体学习的积极性与能动性发挥出来，更好地认知组织学习战略与运营流程，提升组织学习的效果。

组织学习的运营渠道是贯彻组织学习战略的重要场所。一个完善的组织学习流程，必须以战略为指导，从知识的获取、传播、创新、评估等方面展开，环节和环节之间紧密相连。如果有一个方面出现问题，就会导致渠道不畅，影响组织学习的效果，降低企业的综合素质，进而影响企业的可持续竞争力。

（六）组织遗忘

组织遗忘是指组织遗失或忘记组织知识的过程。组织记忆系统储存了大量的知识，但是随着时间的推移或者某些方面的原因，组织会丧失很多的知识，这是与组织记忆相对的组织遗忘在起作用。

组织遗忘主要有两种类型：组织无意遗忘（intentional forgetting）和组织定向遗忘（directed forgetting）。组织的无意遗忘是指组织在没有意识到的情况遗忘了组织知识，其中包括遗忘对组织记忆系统有害的知识和有益的知识。而组织中的定向遗忘，则是指组织有目的地去遗忘组织知识——一般是阻碍组织发展的知识。因此，组织无意遗忘对组织有益的知识，成为组织学习的重要障碍。这会导致组织资源的浪费，降低组织工作的效率。这为组织提出了重要的警示：组织必须完善组织记忆系统，建设储存和管理知识的知识库，为组织进行有效学习，提升组织工作绩效提供可靠的保障。

本章小结：

1. 知识管理是指组织借助于一定的媒介或其他信息传递系统，对知识进行有效地识别、获取、储存、传播、开发和应用，以提高个人或组织创新价值的能力，从而提升组织绩效和核心竞争力的过程。

2. 知识管理应包含几方面的要素：创设有利于知识管理的文化；建设学习型组织；知识编码与构建知识库；建立知识管理的评估体系。

3. 组织学习是组织以系统思考和组织文化的共同愿景为工具，以组织结构为依托，以技术为保障，在持续进行的内外部信息交流中，重新设计以适应不断变化的环境的过程。

4. 组织学习有多种方法。有多种模型试图揭示组织学习的过程和规律。

5. 组织智障主要包括结构智障、文化智障、管理障碍、心智智障、执行智障、组织遗忘几方面的内容。

本章思考题：

1. 简述知识及知识管理的概念。
2. 知识管理的有效方法是什么？
3. 简述虚拟企业知识管理的理论模型。
4. 试述知识管理的相关理论。
5. 简述组织学习及组织学习的概念。
6. 联系实际谈谈组织学习的过程模型。
7. 常见的组织学习智障有哪几种？
8. 简述组织遗忘的概念及常见的类型，举例说明如何管理组织遗忘。

案例分析：

K-R 是如何进行学习？

Knight-Ridder（以下简称 K-R）是美国第二大报业集团，总部设在佛罗里达。它在全美总共发行 29 种报纸，其中有著名的《迈阿密先驱报》、《费城时报》。K-R 的报纸曾荣获过 62 次普利策奖，赢得了极高的声誉。K-R 在大多数人眼中是一个保守、稳健的企业，管理层中的大多数人都已经在公司工作了多年。十多年来，K-R 报纸的发行量基本保持不变，销售收入稳步上升。但是实际上，K-R 的决策者很早就已经注意到了全球报业的停滞不前甚至衰退，因为尽管人口总在增加，但看报纸的人却有缓慢减少的趋势。为此，K-R 在 1988 年实行了两项重要的举措：（1）收购了 Dialog 信息服务公司，使自己从原来的报业大王一越成为集有线电视、报纸、电脑为一身的多种媒体的信息服务商；（2）执行了“25/43 计划”，对其属下的报纸风格进行了大胆的改革。

是什么使 K-R 这个稳健型的企业做出了如此重大的决策呢?我们从组织学习的各个阶段认识 K-R 产生这两大变革的历程。

发现阶段：公司的总裁 James 在最初阶段起了很大作用。他在公司发展良好的同时发现了报业萎缩的一些信号。为了使公司的人都能认识到这一点——也就是为了使个人的知识渗透到组织层面——James 首先启动了一项以读者为中心的活动，James 还亲自走访了下属的每一家报社，将“一切为读者考虑"的信条灌输到公司的各部分。在 James 的不断重申下，K-R 的大多数管理者与员工都认识到了读者减少的事实和为顾客服务的重要性，并希望付诸行动给予改进。

发明阶段：组织对情况有了深刻认识并开始急切地寻找解决方案。“25/43 计划”最初源自管理者们的头脑风暴，然后又收集了大量数据寻找验证。所谓“25/43 计划”是指，它的目标是吸引年龄在 25 到 43 岁的读者群，这些人是战后出生率高峰时成长起来的新一代，有着和前一辈人极不同的爱好和价值观。数据显示，这些人越来越少看报纸，而是将注意力转向了其他信息媒体：电视、多媒体计算机等。要重新吸引这些人对报纸的注意，K-R 必须彻底改变报纸的内容和风格。为此，公司首先选择了一份很小的报纸 Boca-Roton，在当地进行了改版试验。

公司的另一项计划是向其他信息服务媒体的战略转移。同“25/43 计划”类似，公司先后收购了几家小的有线电视和信息服务公司作为尝试，其中有几次是失败的，但对公司影响很小，公司还从失败的项目中获得了很多有用的经验。

实施阶段：“25/43 计划”从公司的一家报社推广到另一家，在企业内全面展开，报纸的版面设计、色彩、内容都有大幅变动。K-R 花费三亿五千万美元收购了 Dialog 公司，一跃成为信息服务商中的带头者。Dialog 的在线信息服务提供了全球最大的全文信息库，吸引了一万多家出版商和专利文献所有者。Dialog 被全球的企业、学者和专业信息收集者使用。这次收购是 K-R 历史上最大的一次，使其自身规模扩大了一倍，业务扩展到 8 个国家。K-R 一夜之间就从在线信息服务业的边缘跳入了这一行业。

推广阶段：两个项目的实施使 K-R 在组织行为和观念上发生了巨大的变化。在组织观

念上，组织成员对自己的企业有了新的认识：K-R 不再仅仅是一个报业的大亨，而是一个著名的信息提供商、一个国际化的组织。此外，员工对信息在现实世界中的形式和提供方式也有了崭新的认识，拓宽了经营的思路。K-R 的长期目标演变为：提供读者需要的任何信息——不论何时何地或以何种方式。组织对企业成功因素的认识也发生了改变。原来大家认为“高质量的报纸+不断增长的人口=不断增长的销售量与广告收入”，而现在新的技术和变革使他们认识到，信息内容成为成功的关键因素——而不是信息的媒介，那些拥有和传播重要内容的企业将会成功。在组织行为上，倾听读者的要求成为新的准则。

反馈阶段：以上各阶段都带给决策者有益的反馈，在今后组织的进一步发展与变革中发挥了作用。

知识库：显然，在两个项目的设计和执行过程中，组织积累了很多的知识。在第一阶段，K-R 的组织成员在头脑中都有了“为读者服务很重要”的观念(知识)，所以，后来在发明阶段所规划的策略就反映了这一点。实施阶段为企业提供了很多市场、经营战略、多元化经营方面的经验与教训。推广过程中，新的组织观念存入了“知识库”，成为组织知识的积累。所有这些知识都以隐性或显性的形式留在了组织的内部。

值得一提的是，在这次学习过程中，K-R 采用了几项避免组织学习智障的措施：（1）大量的数据收集和数据挖掘;（2）外部同盟的建立:同银行、咨询公司建立联系，听取他们的观点，每次开会讨论时都请外部人员参加，广泛吸取意见;（3）无等级制度的决策过程:所有高低层员工的意见被同样重视。

问题：

1．从本案例中你得到了哪些启示？

2．“K-R”的做法还有哪些地方值得改进？

第九章 组织中的决策

1912 年 4 月 14 日，号称“不沉之城”的泰坦尼克号载着 2224 名乘客和船员撞上冰山，1513 人葬身冰海。事故分析显示，尽管当时有 4 条信息显示前方可能有冰山，而且一名瞭望员提出要借助双筒望远镜，但根本未被理睬。船长爱德华·史密斯是一名支配型又受尊敬的领导者，命令船只在黑夜中全速前进。一种无懈可击的错觉告诉人们：“就连上帝本人也无法使这艘船沉没。” 在群体压力下，船员们斥责那名不能使用肉眼的瞭望员，这名瞭望员抛弃了他的疑虑。另外，泰坦尼克号上的电报员也没能把最后也是最完整的一条冰山警报传达给船长。

泰坦尼克号的悲剧在一定程度上是群体决策的后果。在这一决策中，出现了严重妨碍群体决策有效性的不利因素。在本章中，笔者将分析组织中的决策过程，并探讨决策偏差和障碍问题。

第一节 决策过程

决策通常被定义为从多个备选方案中做出选择的过程。它是组织中最重要的管理活动之一。作为管理者，需要做出的决策大到决定组织目标、所提供的产品和服务、财务运作方式，小到生产日程安排、招聘新员工等。这些决定如何做出，有什么方法可以改进决策呢？

一、决策的基本步骤

多数研究者将决策分解为一系列阶段，用于描述个人或群体的决策过程。获得 1978 年诺贝尔经济学奖的赫伯特·西蒙（H.Simon）是研究人类判断和决策过程的心理学家，他曾将决策过程分为三个阶段：第一阶段从事智力活动，对环境进行搜索，确定决策情境；第二阶段从事设计活动，探索、发展和分析可能发生的行为序列；第三阶段从事选择活动，在可能的行为序列中选择一个。

后续的决策研究者分别提出了各自的决策阶段划分（例如 Mintzberg, 1976; Harrison, 1987, 1999），这些阶段划分对理解组织决策的本质特点很有帮助。一个较为全面的模型将决策过程分为八个基本阶段，其中前六个阶段属于决策规划，即理解问题和做决定的过程；后两个阶段属于决策实施，即执行所做决策的过程（Wedley & Field, 1983; Nutt, 1993）。需要注意的是，并非所有决策都遵循全部 8 个阶段的模式，有的阶段可能被跳过，有的阶段可能联合进行，有的阶段可能会反复进行。这 8 个阶段包括：

（1）问题识别。决策经常是对问题做出回应，识别问题则是它的第一步，即发现问题

的存在。例如，管理者发现公司不足以支付员工的薪水，这时他认识到问题的存在，需要做出决策。识别问题并不像表面上看这么简单。研究发现，人们经常歪曲、遗漏、忽视或低估某些周围的信息，而这些信息提供了有关问题存在的重要线索。第七章中曾经讲到，人们并不总是准确地知觉社会情境，而且如果认识问题真相会让自己不安，就会在潜意识里忽略这些问题。否定问题的存在是有效应对它的第一个障碍。很多不良决策都是由决策者忽视问题或者定义了一个错误问题而造成的。

（2）定义目标。在认识到问题存在之后，需要确定解决问题所要达到的目标。例如，如果问题被定义为没有足够的现金，那么目标就是增加可用的现金储备。任何可能达到这一目标的方法都应该考虑。

（3）进行预决策。预决策是关于如何做决策的决定。当分析了存在的问题和情境之后，管理者可能选择自己独自决策，也可能委托他人决策，还有可能让群体决策。进行预决策时，应该注意不同的情境需要不同的决策方式。管理者经常依赖自己的直觉或经验来做预决策。但是，近年来已经开发出计算机程序来帮助人们做预决策。这种决策支持系统（DSS）主要基于现有的社会科学研究结果，对人们的决策过程很有帮助，尤其是可以帮助人们产生大量备选解决方案。

（4）开发备选方案。在识别可能的解决方案时，人们主要依赖于过去曾经有效的方法。例如，解决资金短缺问题的可能方案包括削减劳动力、出售不必要的设备和材料，或者增加销售。

（5）评估备选方案。备选方案一旦确定，就需要对它们进行评估和比较，力图选出最优的方案。在可选方案中，有些可能有效性更高，有些可能实施起来比较困难。例如，增加销售可以增加收入，但实施起来并不那么容易，它只是一种可能的方案，并不是马上就可以实行的。

（6）做出选择。当评估了多个选项之后，选择一个可以接受的方案。尽管理性决策模型假定人们会选择最优的方案，但我们将会看到，现实中的决策往往由于各种条件限制，只会选择可以接受或合理的解决方案，并不一定是最佳的。

（7）执行选择。做出选择之后，实施选中的方案。

（8）追踪调查。检验决策实施的有效性对组织来说非常重要。在决策实施之后，问题是否解决？有没有产生新的问题？也就是说，必须积极寻找解决方案有效性的反馈信息。因此，决策是一个循环往复的过程，如果问题没有解决，那么决策过程还要进行下去。

另一个被经常用到的决策过程模型是哈里森（Harrison）的决策模型。他把做出选择作为决策过程的最后一步，将决策分为 6 个阶段：第一，界定问题。当事件的理想状态与现实状态存在差距时，问题便出现了。第二，确定决策标准，即确定选项优劣的标准。第三，给标准分配权重。并非所有标准都同等重要，需要排列它们的优先顺序。第四，开发备选方案，即列出所有可能的解决方案。第五，评估备选方案。将备选方案根据阶段二和三确定的标准进行评估。第六，选择最佳方案。计算出在各项标准上总分最高的方案，选择这一方案。

上述每种决策模型都明确界定了各个阶段，并从第一阶段开始，一直到最后一个阶段。但需要注意的是，决策是一个动态的过程，几乎每个阶段都可能会发生跳过、停留或重复等现象。并且，下文将会提到，现实中的决策往往并非完全理性，不一定会严格遵循每一

阶段直至选出最佳解决方法。尽管如此，这些决策模型还是提供了分析组织中复杂决策过程的有效工具。

二、决策的个体差异和文化差异

（一）决策的个体差异

个体做决策的方式并不一样，有人确定解决方案的时间比别人长，有人更乐于冒险，有人更关注决定留给他人的印象，有人强调不惜任何代价的成功，有人更依赖逻辑和分析，有人更有直觉和创造力等等。研究发现，人们的决策风格有所不同。这些风格反映了人们如何认知周围世界、如何处理信息等方面的区别。有人提出了区分人的决策风格的两个维度：思维方式和个体对模糊的耐受性。在思维方式维度上，有的人是理性和逻辑的，对信息进行序列加工；另一些人是直觉和创造的，从整体上认知事物。在个体对模糊的耐受性维度上，模糊耐受性低的人需要结构化信息，从而使模糊性降到最低水平；而模糊耐受性高的个体则可以在不确定的环境中工作，在同一时间进行多种思考。这两个维度的交叉，就形成了图 9.1 所示的 4 种主要的决策风格。

模糊耐受性 \ 思维方式	理性	直觉
高	分析型	概念型
低	指导型	行为型

图 9.1　决策风格模型思维方式

分析型的个体对模糊的耐受性高，希望得到更多的信息，考虑更多的备选方案，同时使用更多的时间进行决策。分析型的管理者属于认真的决策者，能够适应和处理新的、意想不到的情境。

指导型的个体难以容忍模糊性，并且寻求理性。这种决策者有效率和有逻辑，但是他们对效率的关注导致收集很少的信息和评估少量的选项之后就做出决定。指导型的个体迅速做出决定，并关注短期效果。

概念型的个体倾向于使用来自多个来源的数据，考虑很多备选项，并根据直觉进行决策。他们关注长期效果，善于发现创新的问题解决方案。

行为型的个体则关注组织中的人们和他们的发展。他们关心下属的幸福感，愿意接受他人的建议。他们倾向着眼于短期效果，并且在决策中对数据的使用不予重视。这类管理者努力避免冲突和寻求被接纳。

研究表明，多数管理者并非只属于一种类型。通常管理者存在两种或更多不同的决策风格，一种占主导，其他是辅助。一些管理者几乎完全依赖于主导风格，而更灵活的管理者则可以根据情境进行调整。但通常个体的主导风格影响很大，不同主导决策风格的个体

之间可能会发生冲突。例如，高度指导型的管理者可能很难接受一名缓慢的、谨慎的分析型风格的下属。研究者发展了决策风格量表，用于测量个体的决策风格的相对强度。个体在某一决策风格上的分数越高，这一风格主导该个体决策的可能性就越大。使用决策风格量表的研究得出了有价值的结果。例如，若公司总裁在每一类型上的得分几乎是一样的，这意味着他们没有主导风格，而是能够在不同类型间来回转换。而不同群体的主导决策风格也可能是不一样的。例如，军队领导并不像通常所认为的那样高度专制，而是有较高的概念型风格得分。这些不同的风格各有其优缺点。例如，指导型的决策者通常快速决策，但往往比较独裁；概念型的决策者愿意冒险、喜欢创新，但决策时往往犹豫不决。人们的人格及人际技能的不同，造成他们以不同的方式进行决策。另外，决策有一定的性别差异。研究发现，女性在决策时分析问题的时间更长，她们往往在决策前过度分析问题，决策后又反复讨论已做的决定。积极一面是对问题的细致考虑，但消极一面是造成决策难以做出，以及对过去决策的后悔。对决策风格的分析除了提供理解个体差异的框架，还可以解释有相同智商的管理者，在分析了相同的信息之后，为什么决策过程和最终的选择完全不同。认识人们的决策风格还可以帮助理解组织中的社会交往。

（二）决策的文化差异

尽管不同文化背景下遵循的基本决策步骤差别不大，但是决策的方式有较大差异。决策者的文化背景影响其对问题的选择和分析的深度、对逻辑和理性的重视程度，以及组织决策由个体管理者独断做出还是由群体共同做出等。首先，不同文化的时间观念有较大差异。如果一名美国人正在管理一个大型建筑项目，他发现最重要的供应商之一交付必需材料的时间要推迟几个月，这名美国人很可能会考虑更换一个供应商。但如果是在泰国、印度尼西亚或马来西亚，管理者很可能会接受这一点，允许项目推迟。也就是说，对于美国、加拿大或西欧的管理者来说，这种情境会被知觉为需要决策的问题，但泰国、印度尼西亚或马来西亚的管理者根本就不会认为这是问题。由于决策过程开始于问题识别，所以必须注意的是，并非所有人都把同样的情境知觉为问题。另一个文化差异也与时间有关，表现在决策所花费的时间上。在美国，好的决策者的标志之一是他很果断，也就是说乐意做出重要决策并且毫不迟疑。但是，在其他一些文化背景下，时间的紧迫感不被重视。例如，在埃及，事情越重要，决策者就被期望花越多的时间去考虑。在整个中东，迅速做出决定会被认为过度草率。

不同文化之间强调的决策主体也有所不同。例如，在美国，人们高度重视个人主义，通常是个体决策。但是，在集体主义文化下，例如日本，如果不先赢得同事的同意，就自己做出决定会被认为不可思议。什么人做决策也有文化差异。例如，在瑞典，决策可能涉及到的所有员工都要参与决策。但在印度，期待的是专制决策，如果管理者咨询下属的意见，会被认为是软弱的表现。

三、组织中决策的类型

组织中的决策有很多类型，根据决策的常规性、包含的风险以及由组织中的什么人来

做决定，可以做以下区分：

（一）程序化决策与非程序化决策

有些决策是根据一套预先设定的选项，重复做出的。例如，一个文档操作员决定将每天的工作都在磁盘上做一份拷贝，一名快餐店经理在供给开始下降时决定订购汉堡包用的小圆面包。这类决策属于程序化决策（programmed decisions），即常规决策，由低层人员做出，依赖于预先确定的行动惯例。而非程序化决策（nonprogrammed decisions）没有现成的解决方案。决策者面临一种独特的情境，解决方案是新颖的。例如，一名试图找到罕见疾病治疗方案的学者面临的是一个没有客观标准的问题，必须依赖于创造性而不是先前存在的答案，才能解决手头的问题。

某些类型的非程序化决策叫做战略决策（strategic decisions）。这类决策通常是由高层主管联合做出，对组织具有重要的长期意义。战略决策反映了以特定方式指导组织的一致模式，需要根据基本的组织价值观或使命进行决策。例如，一个组织可能会做出以特定速度增长的战略决策，或者由某种公司道德规范指引的战略决策。这些决策指引组织未来的方向，因此都可以被看作战略决策。

程序化决策和非程序化决策主要有三个方面的不同：首先，要完成的任务不同。前者的任务是普通的和常规的，而后者则涉及独特和新颖的任务。其次，对组织政策的依赖程度不同。在进行程序化决策时，决策者可以依靠来自组织政策和程序的指导方针。而非程序化决策需要使用创造性解决方案，过去的方案很少有帮助。最后，做决策的人员不同。前者通常委托低层员工完成，而后者则由高层管理人员做出。

（二）确定性决策与不确定性决策

如果可以知道未来会发生什么，那么做决策是很容易的。遗憾的是，人们从来没法准确知道未来，只是有些时候更确定一些。与决策相关的因素的确定性是组织决策者所渴望的。确定和不确定的程度用风险这一概念来表示。所有的组织决策都包含某种程度的风险，从完全确定（没有风险）到完全不确定（高风险）。为了尽量把决策做到最好，人们寻求各种方法来管理所冒的风险，即通过掌握与决策相关的信息，把一个决策所带来的风险减到最小。

风险常常用渴望的结果出现的概率来表示。决策者试图确定当其他事件存在时，某种特定事件发生的可能性或概率。例如，一名金融分析师可能会报告，当最低贷款利率下降时，某一股票有80%的可能上涨；一名气象学者可能报告降水概率是50%。这些数据属于客观概率（objective probabilities），因为它们基于具体的、可验证的数据。还有许多决策是基于主观概率（subjective probabilities），即对何事会发生的个人信念或预感。例如，一名赌徒因为一匹马的名字与他孩子的名字类似而在它身上下注，一个人因为他刚洗了车就认为不会下雨，这都是根据主观概率所做的判断。

在决策中，不确定性是不受欢迎的。组织中的很多决策者都在试图减少不确定性，从而做出更好决定。那么，当组织遇到高度不确定的情境，不知道将来会怎样时，他们会如何反应呢？研究表明，通过与其他组织建立联系，可以减少决策的不确定性。一个组织对

另一个组织将怎么做知道得越多，它在做决策时的确定性就越高。由于这个原因，决策者对不确定性的反应通常是，降低所在商业环境中其他组织的不可预测性。一般而言，信息可以减少决策情境的不确定性。关于过去和当前的知识可以用于对未来进行规划。计算机技术很大程度上提高了管理者迅速决策的能力，因为管理者可以使用最准确和全面的信息。在线信息服务可以提供与决策有关的最新信息。很多管理决策还基于管理者过去的经验和直觉。也就是说，在做决定时，人们经常依赖于过去有效的方案。这一策略经常成功的原因在于，在某些领域有专业知识的个体知道哪些信息是最重要的，并且知道如何解释这些信息来帮助做出决定。正因为如此，在做重要决定时，人们常常会求助于有经验的专家。这些专家不仅拥有评估各个选项风险所需要的信息，并且知道如何降低风险。

（三）组织管理严密的决策和充分授权的决策

在组织中，传统的做法是除了最琐碎的决策之外，所有的决策都由管理者做出。下属收集信息并上报给领导，领导利用这些信息来做决策。这种组织管理严密的决策又被称为自上而下的决策（top-down decision making），即决策权在管理者手中，低层职员很少或没有做决策的机会。

但是，现在很多组织开始改变这一做法而采用充分授权的决策（empowered decision making）。授权决策也叫参与决策（participative decision making），是指把决策权交给那些可能受决策影响的人或让他们参与到决策过程中来。它允许低层职员对工作相关的问题独立做出决策，而不必征得上级的同意。这一方法的原理是，人们对自己的工作了解最清楚，知道什么决定是最好的。并且，当人们得到授权自己做出决策时，他们更愿意执行该决策，并接受这些决策的后果。不仅可以授权给单个人，还可以授权给工作团队。很多公司实行高度授权的自我管理小组（self-managed teams）制度，让他们自己来做决策。对决策的参与可以提高员工的自主感，让他们更多地体会到工作的意义和乐趣，提高其对工作的满意度，从而更好地发挥创造性。一些研究发现，参与程度和生产效率之间存在正相关。

参与决策的形式非常多样化：可以是所有与决策有关的人员全部参与，也可以部分人参与；可以参与决策过程的所有阶段，也可以是只参与部分阶段；可以是短期的，也可以是长期的等。不同形式的参与对生产效率和员工满意度的影响是不一样的。在实际决策过程中，参与的程度受到一些因素的影响，通常个体或团队越有经验、任务越开放和非结构化，需要参与的程度就越高。

第二节　决策模型

一、理性假设和标准决策模型

传统的决策模型是建立在理性假设基础之上的，这种假设来源于古典经济学理论，认

为人是完全理性的，有能力做出最优化决策。在此基础上，衍生出了一系列的具体假设：一是问题清晰，即假定研究者对决策情境拥有完整全面的信息；二是所有选项已知，假定决策者可以确定所有的标准，并列出所有的可行方案，并且知道所有可行方案的可能结果；三是偏好明确，假定决策标准和备选方案的价值可以量化和排序，以反映其重要性；四是偏好稳定，假定具体的决策标准是恒定的，分配给它们的权重也是稳定的，不随时间变化；五是没有时间和其他条件的限制，假定可以获得相关标准和备选方案的丰富信息。

在理性假设基础上建立的决策模型被称为标准决策模型，其中比较著名的是约翰·冯·诺依曼（John von Neumann）和奥斯卡·摩根斯坦（Oskar Morgenstern）于 1947 年提出的期望效用模型。该模型提供了一套明确的公理，包括有序性、占优性、相消性、可传递性、连续性、恒定性等。根据这些公理和公式可以准确地计算出各类决策的数学预测结果。

曾经有很多管理者将这种决策方式看作最佳的管理决策方法。但是在现实情境中，决策者往往面临各种各样的限制，不可能获得所有需要的信息，偏好不那么明确和稳定，导致真实的决策并不吻合要求结果最优化的标准化决策模型。因此，多数研究者并不认为标准决策模型是对人类实际决策行为的描述。只有当决策者是没有一点经验的新手时，当决策者面临只有少数几个行动选项的简单问题时，当搜索和评估备选方案的成本很低时，标准决策模型才能勉为其难地说明和解释人们的决策行为。遗憾的是，现实生活中的决策很少属于这些情况。

二、有限理性假设和描述决策模型

回忆一下你自己选择大学时的体验，你是否考虑了所有可能的备选方案？是否列出了所有的评价标准？是否根据这些标准评估了所有的备选方案，从而找到最合适的学校？多数人的回答是“否”，也就是说，不太可能进行完全理性的决策。同样，组织中的决策也是这样。例如，假设人力资源经理要招聘一名新的接待员，在面试了几名应聘者之后，他可能会选择会见过的人中最好的那名，并停止面试。这名经理不可能去面试所有可能的候选人，然后决定最佳的人选。在找到一名足够好的人选之后，他便停止寻找。

上面的两个例子很好地说明了现实生活中的决策是如何做出的。为了很好地解释这种现实的决策过程，诺贝尔奖获得者赫伯特·西蒙提出了有限理性假设。该假设认为，现实世界充满不确定性，人类的理性资源和认知能力是有限的，不可能掌握全部的信息并选出最佳的选项。因此，决策者并不是考虑所有可能的选择 ，而是考虑那些可用的选项，并选择第一个满足可以接受的标准的选项。也就是说，决策者通常选择的是足够好的，而不一定是最好的。西蒙指出：“同人类生活环境的复杂性相比，人类的思维能力非常有限。”因此，人类必须愿意“找到问题的‘足够好的’解决方法以及行动的‘足够好的’路线”（Simon, 1979）。实际上，人们做决策时，往往追求的是满意（能够满足需要的方案）而不是最优（效用最大化的方案）。

在有限理性假设基础上派生出来的决策模型被称为描述决策模型或行为决策模型，即试图描述人类真实决策过程或行为的模型。西蒙曾试图从以下几方面描述管理决策行为：

（1）决策者从备选方案中进行选择的时候，力图寻找令人满意的方案，满意的标准可以是足够的利润或市场份额，以及公平的价格等。

（2）决策者建构真实世界的简化模型，从问题中抽取重要的特点，而不是抓住问题的所有复杂方面。然后，个体可以在简化模型的范围内进行理性行为。

（3）由于决策者遵循满意而非最优化的原则，他们在进行选择时不需要首先确定所有可能的方案，也不用保证这些方案已经涵盖了所有的可能性。

（4）决策者使用相对简单的经验原则，或者根据习惯来进行决策，这些技术不会对思维能力要求过高。

西蒙的有限理性假设和描述决策模型比完全的理性模型更好地描述了人们实际是如何做决策的。但是，这个模型并没有说明决策偏差如何产生，也不能具体预测人们面临多个选项时到底会选择哪一个。西蒙之后最广为接受的模型是丹尼尔·卡尼曼（Daniel Kahneman）和阿莫斯·特韦斯基（Amos Tversky）于 1979 年共同提出的前景理论。

三、前景理论

卡尼曼和特韦斯基提出的前景理论（prospect theory）解释了人们如何评估备选项以及如何做出决定。前景理论的要点可以用图 9.2 来表示。图中横坐标表示事物的客观状态，右边是正值（收益），左边是负值（损失），它们可以是金钱的赚赔、工作的得失等。纵坐标表示主观状态或对事物变化的心理反应，也叫价值。图 9.2 的含义可以总结为三个要点：

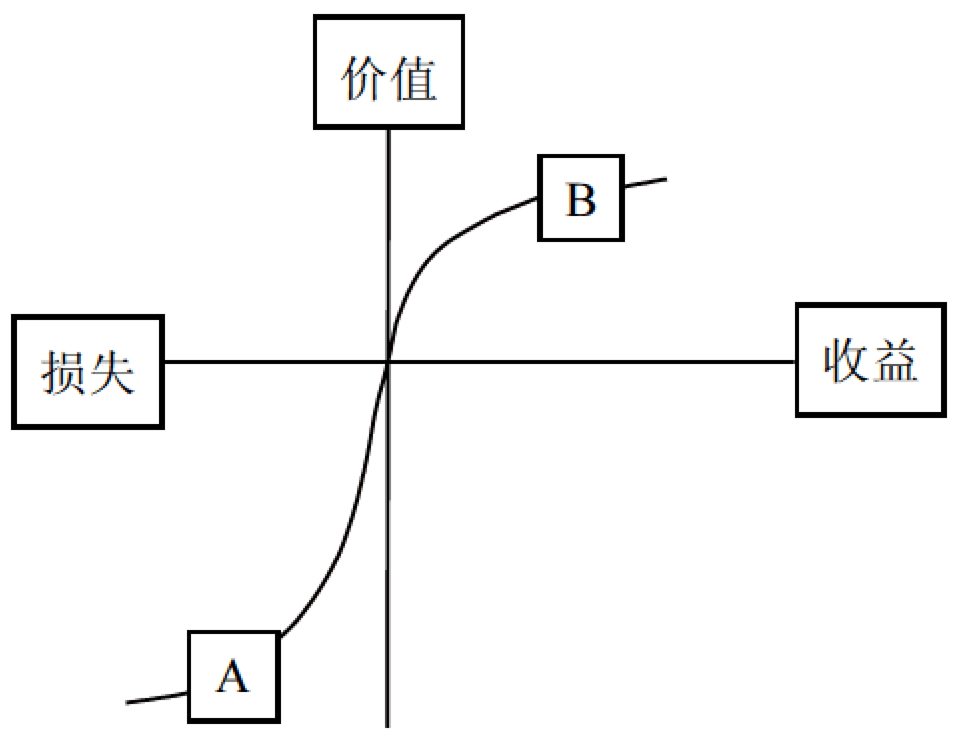


图 9.2　前景理论

（一）价值函数呈“S”型

从图 9.2 可见，收益和损失的价值函数都是曲线，表明客观事物引起的心理价值感并不是与客观状态相对应。对应于收益的价值曲线表示人们对正面事件的反应，它随着曲线

向右延伸倾斜程度逐渐降低，说明随着获利额的增加，人们从每个单位获得的满足感逐渐降低。也就是说，获得 200 元钱带来的满足感不到获得 100 元钱带来的满足感的两倍。因此，如果要求被试从稳赚 100 元钱和 50%的概率赚 200 元钱中选择一项，人们将会选前者。卡尼曼和特韦斯基指出，在面临收益时，人们是回避风险的。而对应于损失的价值函数也是曲线，说明随着损失金额的增加，每单位损失造成的损失感逐渐减少。也就是说，损失 200 元钱造成的损失感不到损失 100 元钱造成的损失感的两倍。因此，如果要求被试从稳赔 100 元钱和 50%的概率赔 200 元钱中选择一项，人们将会选后者。正如卡尼曼和特韦斯基所指出的，在面临损失时，人们是趋近风险的。

（二）损失曲线比收益曲线陡峭

由于损失的价值函数比收益的价值函数更陡峭，造成损失比收益显得更突出。也就是说，损失 100 元钱的负面感受比获得 100 元钱的正面感受要更强烈一些。有研究者认为，同样的损失和收益，前者对心理的冲击比后者要大一倍以上。因此，人们有回避损失的趋势。夸特龙（Quattrone）和特韦斯基（Tversky）指出，这种回避损失的倾向会使谈判变得更复杂，因为双方都将自己的让步看成一种损失，而损失比让步带来的好处显得更突出。回避损失所产生的另一种结果是，当人们拥有一件物品时，它的价值就增加了。例如，当要求人们对自己拥有的一件东西定价时，人们给出的定价通常比他们为购买这件物品而愿意出的买价更高。这一效应之所以出现，就是由于损失的感觉比获得同一件东西的感觉更强烈。这一效应也被很多公司加以利用，这些公司经常向顾客提供退款保证，诱使他们做出购买决定。而人们一旦拥有产品之后，放弃产品就意味着损失，因此多数人不会退还。

（三）中心点的位置

前景理论认为，问题描述上很小的变动就可以改变中心参照点的位置，从而改变人们有关损失和收益的想法，并最终影响决策。如果相对某一参照点，某项结果看来是一种收益，那么决策者会倾向于回避风险（risk-aversive），选择比较保守的那个选项；而如果相对另一参照点，某项结果看来是一种损失，那么决策者会倾向于趋近风险（risk-seeking），选择比较冒险的那个选项。因此，对于各个选项的描述方式会很大程度上影响人们的决策，卡尼曼和特韦斯基称之为框架效应（framing effect）。

前景理论可以很好地对人们的决策行为和结果进行预测，是一个有用的描述决策模型。除此之外，卡尼曼和特韦斯基还研究了人们决策时所使用的方法。他们认为，当面临时间有限、信息不完全等不确定的决策情境时，人们依赖于启发式（heuristics）而不是正规的分析方法。启发式是能够提供捷径的一些非正式的经验法则，它们能够降低做决策的复杂性。卡尼曼和特韦斯基提出的启发式包括代表性启发式（representative heuristic）、易得性启发式（availability heuristic）、锚定和调整启发式（anchoring and adjustment heuristic）等。这些启发式可以帮助人们用很少的时间和努力达到令人满意的结果，甚至很接近标准决策模型确定的最优方案。但是，在某些情况下，启发式的使用可能会导致一些可以预测的系统性偏差。这些偏差的具体阐述见后文。

第三节　个体决策

在组织中，有些决策是以个体为主体做出的，这些个体决策是如何做出的？除了标准决策模型和描述决策模型所阐述的规律外，个体决策还受到一些因素的影响。

一、个体决策中的创造性

创造性是指就特定情境而言，产生新颖和合适想法或产品的能力。这些想法不同于以前用过的，但对当前的问题或机遇又是合适的。创造性可以使决策者充分评估和理解问题，发现他人没发现的问题。并且，创造性最明显的好处是帮助决策者确定所有可行的备选方案。

研究发现，人们的创造性有一定的个体差异：只有不到1%的人具有超常的创造性，10%的人有较高的创造性，60%的人有一定程度的创造性。这说明，多数人都有创造性的潜能，问题是如何在决策中使用它。那么，如何激发员工的创造性呢？

研究表明，三种模式可以提高创造力水平：冒险模式，高创造性的个体更愿意进入未知的领域；准备模式，高创造力的个体在希望有所成就的领域中花费多年获得专业知识；内在动机模式，高创造力的个体能从所从事的工作和创造的事物中得到快乐和满足。

与此相应，研究者提出了一个三成分创造性模型，认为创造性主要需要专业知识、创造性思维技能和任务导向的内在动机。专业知识（expertise）是创造力工作的基础。不管是艺术家还是科学家，他们关于自己领域的丰富知识和理解，是做出创造性贡献的必要条件。当个体具备他们所从事领域中的能力、知识、熟练和专门技能时，他的创造力潜能也有所提高。创造性的思维技能（creative-thinking skills）包括与创造性有关的人格特质、类比能力以及从不同角度看待熟悉事物的才能。研究发现，智力、独立性、自信、冒险、内控、对模糊性的容忍，以及面对挫折时的百折不挠等个体特质与创造性想法的开发有关。类比的有效运用可以使决策者把一种情境中的想法应用到另一种情境中，它们可以使陌生变得熟悉，也可以使熟悉变得陌生。任务导向的内在动机模式是人们愿意从事某项工作的愿望，原因是该工作有趣、使人投入、令人兴奋、使人满足或对个人有挑战性。这个动机成分把创造潜能转化为实际的创造性想法，它决定了个体充分使用其专业知识和创造性技能的程度。因此，创造性高的人们通常热爱他们的工作，甚至达到痴迷的程度。对组织来说，很重要的一点是，个体的工作环境对内在动机有很大影响。工作环境中的一些激励因素可以培养人们的创造性，包括鼓励想法交流、对想法进行公正和建设性的评价；对创造性工作的认可和奖励；充分的财力、物力和信息资源；决定做哪些工作和如何做的自由；有能够有效沟通、对他人表现出信任、支持工作群体的上司；有相互支持和信任的工作群体成员等。

二、个体决策中的理性与直觉

决策的完全理性模型倾向于认为，决策者在面临任何问题时都进行认真推理和分析，经过严密的逻辑思考之后做出决定。但是，正如前文所述，人们的信息加工能力并非无限，不可能所有决策都遵循完全理性的规则。并且，有些时候依赖直觉做出的决策甚至优于理性分析后的决策。

越来越多的心理学研究证实，人们的思维只有一部分是受控制的，即有意识和深思熟虑的，还有很大一部分是自动化的，即无意识和无需努力的。研究者承认，无意识的信息加工可以产生瞬间的直觉。直觉的拥护者认为，重要的信息即使不经过有意识的分析也可以获得。还有一些人支持直觉管理，他们认为，在判断他人的时候，我们应该依赖右脑的非逻辑智慧。例如，聘用或解雇员工时或进行投资时，直觉主义者提倡倾听直觉的呼声。

到底什么是直觉（intuition）？有多种多样的理解。在管理文献中，有人认为直觉的主要属性是速度和决策者不能确定决策是如何做出的；有人认为直觉存在于意识下水平，这就是决策者不能用言辞表达决策如何做出的原因；还有人将直觉决策定义为从经验中提取精华的无意识过程，它不一定脱离理性分析而独立运作，而是与理性分析相辅相成的。其他对直觉的定义包括：迅速有效地了解或识别一个情境的各种可能性的能力（Agor, 1989）；习得行为序列的顺利和自动执行（Isenberg, 1984）；成为固定习惯的分析能力和识别后的快速反应能力（Simon, 1987）等。从上述对直觉的定义，可以概括出以下几点：首先，直觉是很快的；其次，直觉在意识下的水平被利用；第三，直觉包含习得的信息加工模式；第四，直觉在决策中可能是一种积极的力量。因此，决策中的直觉并非神奇的第六感，而是建立在多年的经验基础之上，并开始于对问题的辨认和识别。

在管理实践中，也有证据表明，管理者经常利用直觉来做决策。人们不再武断地认为依赖直觉进行决策是非理性或无效率的，而是在一定程度上开始认可直觉决策。很多人认识到，也许理性分析的力量被过分高估了，在某些情况下，依赖直觉可以提高决策水平。在需要进行快速决策的工作中，例如消防员、特护护士、战斗机驾驶员、特种部队成员等，有经验者所看到的世界与新手有很大差别，而且他们看到的东西告诉他们应该做什么和如何做，决策时主要是直觉在起作用，正式的决策规则很少起作用。下面的例子可以说明直觉是如何起作用的。

消防队长依赖直觉做出正确决定

俄亥俄州克利夫兰的一名消防队长和他的手下面对着一栋后部起火的房子。队长让他的喷水小组进入屋内。他们站在起居室里，向着从厨房中冒出的浓烟和火苗喷水。然而，火势刚被扑灭不久，便再次来袭并继续燃烧。他们再次把火扑灭，但火很快又燃了起来，并且比以前更加猛烈。就在消防队员退出并重新组队进入时，队长忽然有一种十分不详的感觉。他命令所有人撤离现场。就在他的队员刚刚走上街道时，起居室的地板突然塌陷。如果队员仍然呆在房里，他们都会掉进地下室熊熊燃烧的火焰中。

为什么消防队长会做出撤离命令？因为火情情况与他的预期非常不符。由于大部分火焰在起居室的地板下燃烧，因此消防队员的攻势并没有对火情造成太大影响。另外，屋内异常灼热，相比可以看到的火势来说实在是太热了。另一个说明这并非小型的厨房火灾的线索是，从厨房中喷出的火焰听起来声音很小，但灼热的火苗却声音很大。消防队长凭直觉感到地板被下面巨大的火焰所笼罩。

老消防员积累了大量经验，他们可以下意识地对火情进行分类，并以此确定自己的反应。他们寻找情境中的模式或线索，来指导他们一步又一步的行动。

（资料来源：Breen, 2000; Robbins, 2005）

什么时候人们更有可能使用直觉决策？研究者确定了 8 种条件：存在很高的不确定性；很少有先例可以利用；变量很难按科学的方法预测；“事实”很有限；“事实”没有明确指明方向；分析性数据用处不大；有几种似乎合理的备选方案，每种都有很好的根据；时间有限，并且有压力要做出正确决策。

尽管直觉决策已经获得认可，却很难指望人们轻易承认自己在使用直觉决策，尤其是在北美、英国等推崇理性分析的文化中。具有较强直觉能力的人们通常不会告诉同事他们如何得出结论。由于理性分析仍然是更被社会赞许的，直觉能力经常被伪装或隐藏。另一个问题是：能否教会管理者使用他们的直觉？有研究者通过研讨会来开发管理者的直觉技能，在一些组织中取得了积极结果。一篇关于直觉的综述研究表明，尽管直觉本身不能被教会，但可以训练管理者更充分地依赖直觉的提示。

三、个体决策的偏差

在决策模型的讨论中，我们提到，决策者在很大程度上是有限理性的。并且，决策者经常依赖于捷径和经验法则（卡尼曼和特韦斯基称之为启发式）来节省精力和简化决策程序。通常，这些捷径有助于人们迅速做出令人满意的决策。但是，它们也可能会造成严重的偏差和对理性的偏离。因此，认识这些偏差可以有效地帮助决策者，在利用某些经验法则的同时，避免其负面的破坏作用。以下是一些常见的决策偏差。

（一）证真性偏差（confirmation bias）

理性决策过程假定，人们总是客观地收集信息，但实际上并不是这样。人们选择性地收集信息。证真性偏差是选择性知觉的一种特定形式。人们寻求能证实自己过去选择的信息，忽视那些与自己作出的判断和决策相抵触的信息。人们还倾向于接受与自己预先形成的观点相一致的信息，对那些挑战这些观点的信息则持批评和怀疑态度。因此，个体所收集的信息往往偏向于支持自己已经持有的观点。证真性偏差影响人们到哪里或到谁那里去收集证据，因为人们总是希望看到自己想看到的、听到自己想听到的内容。它还造成人们对支持自己观点的信息给予较高权重，而对反对自己观点的信息给予较低权重。

（二）过度自信（overconfidence）

在人们的判断和决策中，过度自信是一个非常普遍的问题，它带来的潜在破坏性也很

可能是最大的。当给人们一些常识性问题，要求回答并估计回答正确的可能性时，人们倾向于过度乐观。研究发现，当人们有 65%至 70%的信心认为自己正确时，实际上正确率只有 50%；当人们说自己 100%确定时，实际的正确率只有 70%至 85%。在主观信心和实际正确率之间只有很小的相关，或者没有相关。

在组织中，一项与过度自信有关的有趣发现是，能力不足会增加过度自信倾向，那些智力和人际能力最差的人最有可能高估他们的成绩和能力。管理者或员工对一个问题掌握的知识越多，他们表现出过度自信的可能越小。当组织成员考虑自己专业领域之外的问题时，过度自信更有可能出现。过度自信的部分原因来自于上述的证真性偏差，人们总想证明自己是正确的，所以就产生了过度自信。要降低过度自信，一种方法是及时获得反馈信息，另一种方法很简单，却更实用和有效，就是经常反省自己的信念出错的原因。

（三）易得性偏差（availability bias）

决策者往往会根据一些容易想起的典型事例做出判断和决策，这就是易得性启发式。很多时候这一启发式可以发挥积极的作用，但在更多情况下会导致系统偏差。很多人担心飞机失事，却不太担心驾车发生事故，他们认为乘飞机更危险。但实际情况并非如此。统计显示，美国旅行者在出行同样距离的情况下，发生汽车事故死亡的概率是飞机失事的 26 倍（美国联邦安全委员会, 1991）。之所以有这样的判断偏差，原因在于媒体对飞机失事的渲染，人们很容易回忆起飞机失事的事例或场景，因而高估了乘飞机的危险。那些刺激情绪的、格外生动的或者最近发生的事件更容易从记忆中提取出来，因而人们高估这些事件的发生概率。从事说服教育工作的人经常利用人们的易得性偏差，举出一些典型的事例来影响人。易得性偏差让某些管理人员进行年度绩效评估时，更重视雇员最近的行为，而不是半年以前的表现，从而导致不可靠甚至是不公平的评估。

（四）代表性偏差（representative bias）

代表性启发式是指根据当前刺激或事件与已有范畴或概念的相似程度来进行判断和决策。例如，如果你发现自己的邻居比较传统和害羞，你会如何猜测她的职业？作出此种判断的重要依据是他与生意人、医生、服务员、律师、图书管理员等职业成员的相似程度。她跟哪个职业的典型成员更相似，你就会认为她从事的是那种职业。结果是，你认为她是图书管理员。但是，这种代表性启发式可能会出错。因为这种判断或决策往往忽视了基础比率，即在总人口中特定刺激或事件出现的频率。事实上，生意人是图书管理员的几十倍，因此你的判断很可能是错误的。管理者也可能受到代表性偏差的影响，经常根据过去成功和失败产品的经验，预测新产品成功的可能性。如果来自同一所大学的三名员工业绩很差，管理者可能会预测，来自这所大学的应聘者将不会是好的员工。

（五）锚定偏差（anchoring bias）

锚定偏差是指固着于作为起始点的初始信息的倾向。一旦设定了初始值，人们不能充分调整随后的信息。初始值或起始点可能是由先前的历史问题呈现的方式或者一些随机的信息决定的。由于人们倾向于给最先收到的信息以与其重要性不成比例的权重，因此很容易出现锚定偏差。因此，与后来接受的信息相比，最初的印象、观点、价格和估计等往往

被给予过度重视。

在广告人、管理者、政治家、房地产代理、律师等专业领域中，说服技能是很重要的。因此，他们广泛地利用人们的锚定偏差，一开始就采取一个极端的立场，试图锚定人们的心向或态度。例如，在一个模拟陪审团审判中，原告律师要求一组陪审员判决一千五百万至五千万美元范围内的赔偿；另一组陪审员则被要求判决五千万至一亿五千万美元范围内的赔偿。由于锚定偏差的作用，前者判定的中数金额是一千五百万美元，后者判定的中数金额是五千万美元。在谈判和访谈中，锚定也很重要。当谈判开始时，锚定也发生了。一旦某人给出一个数字，你就很难忽略那个数字。例如，当未来的雇主询问你上一份工作的报酬时，你的回答通常锚定雇主的出价。很多人了解这一点，因此会向上"调整"先前的薪水，希望雇主给我们更高的报酬。锚定也可以扭曲招聘访谈。当访谈应聘者时，你最初获得的信息很可能会锚定你对他的评估，过度地影响你对后来获得信息的解释。作为决策者，你可能会认为自己可以控制这种初始信息的不利影响，但实际操作中人们很难抵御锚定的作用。因此，在做出决策之前，对于任何极端的锚定值，都应该确定一个反方向的相等的锚定值。

（六）事后通偏差（hindsight bias）

事后通偏差是指在事件发生之后，错误地认为自己早已准确预见到这一事件的倾向。当事情发生之后，人们往往认为这样的结果是很明显和不可避免的，却忽略了自己的判断实际上已经受到已知结果的影响。例如，在一场比赛结束后，人们往往认为他们之前已经成功预测了谁将获得冠军，那样的结果是必然的，但在比赛之前让他们做判断的话，他们不会认为自己有这么高的准确性。出现事后通偏差的原因是，人们不能清楚地回忆起或很容易遗忘事件发生之前自己的判断。而且，人们很善于根据后来得到的信息重构过去，高估事先知道的的内容。因此，事后通偏差可能是选择性回忆和重构过去的结果。

事后通偏差使得人们不能从过去学到东西。它使人们高估自己的预见性，并且对未来决策的准确性过度自信。例如，你真实的预测准确率可能只有 40%，但你认为有 90%，你实际上已经变得过度自信了，并且对自己的预测能力坚信不疑。

（七）随机性错误（randomness error）

人们对随机事件的认识存在一定的问题。多数人认为自己可以控制自己的世界和命运。尽管个体确实可以控制一部分事件，但在世界上存在很多随机事件，是人力无法控制的。人们往往试图从随机事件中寻找规律，导致决策出现偏差。

对随机性的知觉常见的问题是相关错觉（illusory correlation）和控制错觉（illusion of control）。当人们期待发现某种重要的联系时，就很容易把随机事件联系起来，知觉到相关，（Ward & Jenkins, 1965）。将随机事件知觉为有联系的倾向还容易造成控制错觉，即认为自己可以预测和控制随机事件。

设想一下股票市场的交易。尽管股票价格的短期变化实际上是随机的，但很大一部分投资者或投资顾问仍然认为他们可以预测股票价格的走向。例如，当向一组被试提供股票价格和趋势的信息时，这些被试认为自己有大约 65%的把握能够预测股票变化的方向。但

实际上，这些个体的预测只有 49%的正确率，跟随便猜测的水平相差无几。许多研究都发现，人们都在仿佛可以预测或控制随机事件一样在采取行动。赌徒和股票交易者往往表现出控制错觉，认为自己可以控制赌博的结果或股票交易，从而经常导致损失。

对随机事件的错误认识造成的最严重的后果之一是，迷信自己想象和构造出来的虚幻模式。这些迷信可能完全是人为的（“我从不在 13 号的星期五做决定”），也可能是来自某种先前被强化的行为模式（“我总是打我的幸运领带去参加重要会议”）。几乎所有人都有一定程度的迷信行为，但是当它影响日常的判断或者使重要决策出现偏差时，这种迷信是有害的。如果走向极端，某些决策者可能会完全被迷信控制，造成他们几乎不可能改变惯例，也不可能客观地加工新的信息。

（八）承诺升级（escalation of commitment）

决策者经常面临的一个难题是，为了获得正面结果，究竟应该放弃一个失败的行动，还是应该增加承诺或投入，以弥补之前的损失。研究表明，当决策过程实际上包括一系列的决策时，决策者有一种逐渐升级承诺的倾向。承诺升级是指，即使有清楚的证据证实一项决策是错误的，仍然固守该承诺。例如，一位男士 4 年来一直和一位女士保持约会关系。尽管他承认两人的关系并不十分和谐，他仍决定要和她结婚。当询问他做出这一决定的原因时，他回答：“我在这段关系中已经投入了太多！”很多证据表明，当个体认为自己需要为某一行动的失败负责时，他们会在这一活动中增加承诺。也就是说，他们在坏事之后投入更多金钱，为的是证明他们最初的决定并没有错，并且避免承认他们犯了错误。承诺升级也适合解释这样的现象，即人们总是试图表现出言行的一致性，哪怕有迹象表明继续兑现先前的承诺是错误的。

承诺升级对管理决策具有重要启示意义。很多重要决策都有可能发生承诺升级现象，包括资本投资、重大研发投资、工厂扩建等项目。理性决策理论认为，在对各种备选方案进行评估时，应该考虑方案的未来收益，而不是考虑“沉没成本”（sunk cost）。所谓沉没成本，顾名思义，即过去投入的不可挽回的成本，通俗地说就是“打了水漂”的成本。但是，实际中的决策者往往不愿将那些成本看作不可挽回的，因此总是希望挽回“沉没”的成本，缅怀过去而非面向未来，于是将错就错，继续投入资源给那个从开始就注定要失败的行动，从而给许多企业带来巨大损失。

造成承诺升级现象的原因很多，主要有：（1）决策者为了自我辩解，认为有必要为过去的行动向自己或他人提供正当的理由；（2）决策行为的一致性压力。行为一致性常常是有效管理者的特点或标志。因此，管理者为了表现自己是有效的，在转向其它行动更有利的情况下，仍然坚持原来的行动以保持行为的一致性。实际上，真正有效的管理者是那些能够区分不同情境的人，他们知道哪种情境下坚持是有利的，哪种情境下坚持是有害的。（3）证明自己是理性决策者的愿望；（4）过度乐观，高估积极结果出现的可能性；（5）控制错觉，认为自己拥有他人所不具有的控制未来的能力；（6）受到损失框架的影响等。

对于组织来说，在一些情况下可以避免承诺的不断升级。首先，当可用于进一步投资的资金有限，且失败的威胁非常明显时，决策者容易终止失败的投资；其次，当决策者能够分散先前失败的责任时，他们也倾向于终止继续投入，也就是说，如果人们认为自己只

是需要对失败负责的众多人之一时，他们不会再主张为先前的失败继续投资；最后，当人们很清楚地得知，投入的总量已经超过了预期的收益时，他们也不会继续投入。

（九）框架效应（framing effect）

问题呈现方式的不同会导致人们做出不同的决策，这种现象叫做框架效应。特韦斯基和卡尼曼发现，当问题呈现的方式强调正面的收益时，人们倾向于回避冒险，寻求确定的结果；而当问题呈现的方式强调潜在的损失时，人们更愿意冒险，以避免确定的损失。不妨考虑下面这个例子："政府正在准备对抗一种罕见的疾病，这场疾病危及 600 人的生命。现在有两套方案，需要从中选择一套。如果实施方案 A，将挽救 200 人的生命；如果实施方案 B，有 1/3 的概率挽救 600 人的生命，2/3 的概率无法挽救任何人。你会选择那个方案？"特韦斯基和卡尼曼发现，当用这一框架呈现问题时，72%的人选择方案 A，也就是说他们选择肯定挽救 200 人的方案，而不是冒 2/3 的风险去挽救 600 人的生命。同样的问题，改变一下结果的陈述方式又怎样呢？"如果实施方案 C，400 人会死亡；如果实施方案 D，有 1/3 的概率没人死亡，2/3 的概率 600 人都会死亡。"特韦斯基和卡尼曼发现，在这种呈现方式下，反而有 78%的人选择方案 D，也就是说愿意冒险赌一赌，而不是接受肯定死 400 人的结果。

框架效应得到了很多研究的证实。认识这一效应的重要意义在于，它告诉我们，框架会在很大程度上影响决策者是回避风险还是趋近风险。如果仅从避免损失的角度来考虑问题，决策者可能做出过于冒险的决定，以避免承受确定的损失。这种决策倾向常常导致失败。要避免这一问题的出现，决策者应该从不同的角度来看待问题，不仅考虑可能带来的损失，还要考虑可能带来的收益。研究表明，对决策问题变换角度加以思考，尽管可能会增加决策时间，但可以带来更好的决策。

四、改进个体决策

管理者可以做些什么来改进他们的决策呢？著名组织行为学专家斯蒂芬 • P. 罗宾斯（Stephen P. Robbins）提出了 5 点建议：

（1）分析情境。调整自己的决策风格来适应你所在国家或地区的文化以及你所在组织的评价和奖励标准。例如，如果你处在一个不太重视理性思考的国家，不要强迫自己遵循理性决策模型，或者试图使你的决策显得理性。类似地，不同组织的风险偏好、决策机制有所不同，调整你的决策风格以确保它与组织文化相协调。

（2）了解偏差。要认识到偏差的存在，然后努力减少它们的影响。减少偏差和错误的方式见后文中罗宾斯所提的建议。

（3）理性分析与直觉有机结合。在决策中，理性分析与直觉并不冲突。将二者有机地结合起来，可以提高个人决策效力。随着管理经验的获得和增加，应该对自己的直觉越来越自信。

（4）不要以为一种决策风格对每项工作都适合。一个组织之内的工作部门和岗位也有

所不同，就如同组织之间存在差异一样。如果决策风格与工作的要求相匹配，那么你决策的有效性将会提高。例如，如果某人的决策风格是指导型的，那么与那些需要迅速采取行动的人们一起工作将很有效率。这种风格可能很适合管理股票经纪人。而分析型风格则可能更适合管理会计人员、市场调查人员和金融分析人员。

（5）努力提高你的创造性。敢于公开自己的观点，努力寻求新的问题解决方案，试着从新的角度看待问题，并善于使用类比。努力消除可能妨碍创造性的障碍。

除此之外，罗宾斯还专门就如何减少偏差和错误提出了几条建议：

（1）专注于目标。没有目标，就无法保持理性，不知道自己需要什么信息，不知道哪些信息有关、哪些无关，很难从选项之间做出选择，甚至会经常对所做出的选择感到后悔。清晰的目标使决策变得更容易，并帮助你排除与你的利益不一致的选项。

（2）寻找违背你的信念的信息。对抗过度自信、证真性偏差和事后通偏差最有效的方法之一是，主动寻找与你的信念和假定相矛盾的信息。当我们能够坦然面对我们可能犯的错误时，就可以挑战自作聪明的倾向。

（3）不要试图从随机事件中寻找和创造意义。受过教育的头脑总是试图寻找因果关系。当事情发生时，人们问为什么。当不能找到原因时，经常就会虚构原因。必须承认，生活中有些事件在个人控制之外。要经常询问自己，周围所发生的事件是否有意义，或者它们仅仅是巧合。不要给巧合编造意义。

（4）增加你的选项。不管你已经识别了多少选项，你最终的选择不可能好于你已选中的选项集合中最好的那个。因此，增加你的决策选项，使用创造性来开发广泛的、多种多样的选项。你产生的备选方案越多，且选项越是多样化，你找到一个好选项的可能性就越大。

需要注意的是，偏差并不意味着绝对的非理性和有害。正如前文所述，它们常常可以帮助我们迅速做出足够好的决策，只是在某些情境下可能导致严重问题。另外，尽管有各种减少偏差的技巧，但这些技巧的一个共性是：看待问题时使用不同的视角。

第四节 群体决策

在现代组织中，群体决策应用的范围很广。组织中重要的商业决策通常是由委员会、研究小组、特别工作组或评估小组做出的。这样做的好处在于：群体中的相互影响和鼓励可以刺激群体成员产生更多的解决方案；群体成员对决策有更高的承诺；群体带给问题解决情境更多的知识和经验等。这是否意味着群体决策一定优于个体决策呢？对这一问题的回答依赖于很多因素，下文将给出答案。

一、群体决策与个体决策

由于群体决策涉及多人，如何达成最终的一致是许多研究者关心的问题。可以对群体

成员的最初观点与最终的组织决策进行比较，从而得知最终的群体决策是根据什么规则得到的。一种规则是多数决定原则（majority-wins rule），即群体支持大多数成员所持的立场。另一种规则是真理决定原则（truth-wins rule），即随着越来越多的成员认识到它的正确性，正确的决策将会出现。还有一种是 2/3 多数原则（two-thirds-majority rule），即得到 2/3 或以上成员支持的决定将被采纳。最后一种是首先转换原则（first-shift rule），即群体倾向于采纳群体成员的观点第一次发生转变时的决策。研究表明，这些决策规则可以说明 80%的组织决策（Stasser & others, 1989）。很多研究着重揭示特定类型任务使用哪种规则。例如，有研究发现，当不存在客观正确的决策时，即决策反映的是一种偏好时，主要采用多数决定原则；而在有一个正确答案的智力型任务上，主要采用真理决定原则。

群体决策有很多优点。首先，将人们召集到一起可以增加做出好决策所需知识和信息的数量，也就是资源共享。其次，群体决策可以产生工作分工，由于有足够多的人员分担工作负担，个体可以仅完成那些最擅长的任务，从而提高决策的质量。第三，群体决策比个体决策更容易被广泛接受。当人们参与到决策过程中时，他们很可能对决策更加理解，并且更愿意执行这一决策。

但群体决策也有很多问题。最明显的缺点是，群体决策比个体决策需要更多时间来获得解决方案。第二个问题是，在重要问题上的观点不一致可能会导致敌意和群体冲突。另外，群体内部存在从众压力，群体成员希望被群体接受的愿望可能会导致不能畅所欲言地表达自己的观点。群体讨论也可能受到少数人（如专制型领导）的控制，从而阻碍对相关解决方案的自由讨论。最后，群体决策还会导致责任不明和责任分散的后果。

群体决策既有优点又有缺点。那么，什么时候群体决策优于个体决策呢？一般地说，在重大的、复杂的、开放性的问题上，既应该、又适合采用群体决策。假如一家公司正在考虑是否应该与另一家公司合并，这就不是任何一个人可以单独做出决定的事情。其重要性使得任何一个人都无力承担决策的后果和责任，其复杂性使得任何一个人（即便是专家）都没有足够的智慧和能力作出明智决策，其开放性使得群体观点的多样性能够得以充分发挥，因此群体可以做得更好。究竟群体是否比个体做得更好，还依赖于几个因素。首先必须考虑群体的成员是谁。好的决策群体通常由具备互补性技能的异质成员所组成。例如，在决定是否合并这类问题上，由律师、会计师、房地产代理人和其他专家组成的群体做出的决策很可能优于由同一领域专家组成的群体。事实上，群体成员提供观点的多样性本身就是群体决策的主要优势，如果做不到这一点，群体决策的优势也就不存在了。当面临需要了解多方面信息的决策时，单个决策者需要大量时间来查阅资料和咨询他人。而如果群体包括来自多个领域的成员，那么寻找信息所需的时间就会大大减少，观点的多样性也能够得到保证。要使群体决策有效率，仅仅有技能的多样性是不够的，成员还必须以开放的、没有敌意的方式交流观点。如果某一个体或整个群体气氛使他人不敢贡献自己的专业知识，那么即便是由异质专家构成的群体，也难以表现出决策上的优势。拥有专业知识和能否使用这些专业知识是两回事。研究表明，资深专家的意见在群体决策中扮演着重要作用。当资深专家的意见被给予充分重视时，群体才能做出好的决策。

相反，在较不重要的、简单的或封闭性的决策任务上，群体决策的优势就不一定能发

挥出来。假如有一个需要做判断的简单问题，它有一个很容易证实的答案。在这种情况下，群体有可能比个体做得好，但那是因为群体中的某个人可能知道正确答案，是他为群体完成了任务，也就是说，群体只是增加了知道正确答案的可能性。但是，如果群体非常大，那么很可能会影响拥有所需知识的那名个体完成任务。事实上，在一些知识竞赛或智力活动上，单独工作的专家常常会超过群体。原因是，在群体中完成此种任务的专家可能被其他人所干扰，还必须说服其他人自己的答案是正确的。

此外，群体决策和个体决策哪个更好还取决于衡量效果的标准。如果考虑的是准确性、创造性和成员对最终方案的接受程度，群体优于个体。

二、群体极化和群体盲思

真正重要的决定很少交给个人，多数是由群体共同做出的，原因在于人们相信群体决策比个体决策更准确，不容易犯个体常犯的错误。如前所述，当满足一些条件时，群体确实优于个体。但是，也有充分的证据显示，有时候群体对各种备选方案的评估会出现问题，决策的质量也会受到影响，甚至会做出有害的决策。危害群体决策的两种现象分别是群体极化（group polarization）和群体盲思（group think）。

人们普遍认为，群体比个体在决策时更为谨慎，对于这一流行看法，一项研究提出了质疑。研究者设计了一些两难决策情境，要求被试对这些虚拟任务应该承担多高的风险做出评定。评定之后，将被试分为约 5 人一组进行讨论，并就每个问题的解决方案达成共识。结果发现，被试在参加群体讨论后，更愿意建议采用高风险方案。Stoner 将这一现象定义为冒险转移（risky shift）。此后，很多研究对此现象进行了考察。他们使用的材料是两难选择问卷，包括 12 个模拟情境，情境中的主人公需要在冒险行动和保守行动之间做出选择。被试的任务是决定当概率达到多少时，他们将建议主人公采取冒险行动。通常先评估个体被试的冒险倾向，然后约 5 名被试组成群体，讨论 12 个情境，并就每个情境的概率选择达成一致意见。个体与群体之间的平均分差值即冒险转移的测量值。一般的发现是，尽管群体讨论通常使人们更倾向于冒险，但在某些两难情境下，群体讨论之后反而更加谨慎了。通常，当全体成员最初的平均倾向偏向于保守时，讨论之后会出现此种保守转移（cautious shift）。后来，研究者将上述两种现象统称为选择转移（choice shift），而更普遍的称呼是群体极化，即群体讨论会加强成员的初始平均倾向，使得群体的观点向更极端的方向转移，也就是说，使冒险者变得更冒险，保守者变得更保守。群体讨论会极化最初的观点。

在商业领域中也存在群体极化现象。例如，一种常见的有害决策叫做“投资过多难以退出”，导致很多企业蒙受巨大经济损失。一项研究要求加拿大商学院学生设想自己必须做出决定：为了挽回在某些可能失败项目上的损失，是否应该追加投资？结果发现，如果是一个全新的项目，没有前期的投入，很少有人会决定投资；如果已经有所投入，72%的人会决定追加投资，即出现了“投资过多难以退出”现象。但是，更严重的是，如果是在群体中做决策，94%的人选择追加投资。

为什么群体会比个体采取更加极端的观点呢？有两种解释：一种解释认为，人们在群

体讨论中接受的说服性信息较多，而且人们提出的论据往往是支持最初立场的，从而使个体更加确信自己最初的观点。第二种解释强调社会比较的作用。个体关注自己观点与他人观点的比较。在讨论中，个体往往发现其他人有相似的态度，为了获得赞赏，人们会把观点表达得更极端。两种解释恰好对应于从众的两种原因，即信息性影响和规范性影响，所以总的来说，是从众导致了群体极化。

另一种危害群体决策的现象是群体盲思。当群体具有很高的凝聚力，并且相对不受外界影响时，群体忠诚和从众压力会导致群体盲思。贾尼斯将群体盲思定义为："由于群体内压力而造成的心理效率、现实检验以及道德判断的退化。"贾尼斯认为，正是群体盲思导致了美国历史上几个重大的不幸事件，如 1941 年珍珠港在毫无防备情况下被偷袭、1964—1967 年的越战不断升级、1961 年的入侵古巴猪湾事件等。在这些事件中，某些重要领导人压制不同意见，忽视那些使决策显得有问题的声音，最终做出了很糟糕的决策。

贾尼斯认为，群体盲思有几个可能的原因：首先，群体盲思在凝聚力高的群体中容易出现，因为这样的群体与外界观点隔离。其次，强有力的领导人和独断的领导作风也容易导致群体盲思。这样的领导人可能会针对问题提出自己的解决方案，并且强烈维护自己提出的方案，在与领导保持一致性的压力之下，群体成员不敢表达自己的反对意见，或者忽视自己的意见或低估自己意见的重要性。

根据贾尼斯的理论，群体盲思有八个主要症状：第一，群体中大多数人有一种自己无懈可击的良好感觉，这种错觉导致过分乐观和过度冒险；第二，对群体固有的道德观念持毫不质疑的态度；第三，群体对来自外界的警示性信息采取忽视态度，或者对其进行合理化；第四，对竞争对手持有刻板印象，认为他们太邪恶不值得与之谈判，或者认为他们太软弱或太愚蠢而不会构成严重威胁；第五，任何不赞同多数人意见的群体成员都会感受到压力；第六，群体成员对自己与表面上一致的群体意见相背离的想法进行过多的自我审查或检讨；第七，形成了群体意见一致的共同错觉；第八，某些群体成员是自封的群体"精神保卫者"，使群体远离任何可能破坏群体自满感的信息。这些症状会阻止群体成员对相反信息及问题的各种可能性进行搜寻和讨论。当领导力主某种立场，而整个群体又排斥异议时，群体盲思就产生了，从而导致错误的决策。

贾尼斯的观点引起了广泛关注，人们用他的理论分析了很多领域失败决策的案例。例如，1986 年 1 月 28 日，挑战者号航天飞机发射 73 秒之后发生爆炸，所有宇航员丧生。对事发过程中关键人物之间谈话的分析表明，决策中出现了群体盲思。事实上，有高级工程师在发射之前就提出警告，认为空气温度太低，会对安全造成威胁。但是，最高决策者对这一警告置之不理，压制这些工程师的反对意见。而且，发射飞船的决定是由管理层集体投票做出的，工程师根本没有投票权。美国国家航空航天局已经有连续 55 次成功发射的纪录，使得决策者过度自信。尽管一些管理者知道飞船外壳可能有问题，他们没有把这些信息传递给其他决策者。本章开头引述的泰坦尼克号的悲剧也表现出了无懈可击的良好感觉、从众压力、精神保卫者等群体盲思症状。对大公司商业决策的分析表明，高层管理群体对市场条件变化的失察导致它们濒临灾难边缘。问题通常表现为高凝聚力群体的成员对群体的决定非常自信，根本不会对这些决定产生任何怀疑。结果是，他们从众于群体，停止个

人的批判性思考。当群体成员彼此非常忠诚时，他们会忽略那些来自其他来源的有用信息，仅仅因为这些信息可能会挑战群体的决定。

贾尼斯认为，避免群体盲思的最好方法之一是，正式任命某位群体成员充当“魔鬼代言人”角色。这名成员的责任就是对多数人的立场公开提出挑战，提出不同的观点。贾尼斯及其同事推荐的预防措施还包括：群体领导应该明确鼓励不同的意见和批评，包括对他们自身观点的批评，扮演公正无私的角色；群体领导应该避免一开始就表明自己的个人偏好；群体应该要求群体外的专家或有资格的同事参与会议，并鼓励他们挑战群体的一致意见；控制群体的规模；将群体分为几个小群体，然后再重组到一起表达不同意见；在实施决定之前，召开“第二次机会会议”，要求群体成员表达任何质疑和提出任何新的建议等。研究表明，这些措施可以有效地提高群体决策质量。一些组织已经开始有意识地采用这些方法。例如，有的公司使用了“魔鬼代言人”方法，改善了群体决策的质量；一些公司的执行官有意在讨论中引发争论，从而使决策的消极方面及早被发现。一些日本企业开会时是职位最低的职员首先发言，然后是职位稍高一点的，依次类推。

也有一些研究者对贾尼斯的群体盲思理论提出了质疑。有证据表明，该模型的某些方面与错误决策的联系大于其他方面。例如，独断的领导方式确实与糟糕的决策有关，但是群体凝聚力高则不一定会导致负面结果。但不管怎样，群体盲思的讨论可以帮助决策者避免群体决策的一些陷阱，从而做出更好的决策。

三、群体决策技术

群体决策虽然有很多优点，但也有不利的方面，甚至可能会导致做出不良决策。因此，人们设计了很多方法来提高群体决策的质量。下面介绍几种改善群体决策的技术。

（一）头脑风暴（brainstorming）

头脑风暴是一种激发尽量多观点的方法，它鼓励群体成员发挥想象力，自由地提出各种想法，而把对这些想法的评价延后去做。实施头脑风暴的几条原则是：（1）随心所欲原则：鼓励提出各种观点，越疯狂越好；（2）无批评原则：不能有批评，禁止对任何观点提出负面评价；（3）尽可能多原则：假定想法越多，出现好主意的可能性越大；（4）骑肩原则：鼓励群体成员对别人提出的观点加以整合和改良。典型的头脑风暴通常是 6～12 人围坐在一起，领导者讲明问题，然后大家自由发言，尽可能多地想出各种解决问题的方法。其目的是建立大量备选方案，而不是做出最终的决定。

研究表明，使用头脑风暴的群体的确比不使用头脑风暴的群体产生了更多的观点。但是，尽管人们通常认为头脑风暴法会比个人单独工作产生更多的想法，研究结果却并不支持这样的推测。很多研究证实，单独工作的人比其在头脑风暴中产生了更多更好的想法。

研究者提出了改进群体头脑风暴的方法。首先，将群体和个体头脑风暴相结合，即先进行群体头脑风暴，再进行个体头脑风暴，比反过来进行和单独使用二者之一的效果要好。第二，让群体成员通过书写进行互动，用写和读来代替说和听。第三，采用电子头脑风暴，

即让群体成员通过联网的计算机匿名交流看法。这三种方法的优点在于，克服了传统头脑风暴中同时只能有一人发言的缺陷，避免了个体思路遭到阻塞，还解除了面对面时的社会顾虑和心理压力。研究表明，匿名的电子头脑风暴产生了比面对面的头脑风暴更多的观点。

（二）名义群体技术（nominal group technique, NGT）

这种技术对决策过程中群体成员的人际沟通进行了限定，所以叫做“名义”群体。它通常包括以下步骤：首先，组成群体，但在进行群体讨论之前，每一成员将自己对问题的想法以书面形式写下来；其次，群体成员挨个表达自己的观点，用简单的语言记录他人所有的陈述，在所有观点都被记录下来之前不允许讨论；第三，群体开始就记录下来的想法展开澄清和评估；最后，群体成员安静和独立地对各种想法进行排序，合计出等级最高的想法，做出最终的决策。

（三）德尔斐技术（Delphi technique）

德尔斐技术最早是由 Rand 公司开发的，是一种系统收集和组织多个专家观点并最终做出决策的方式。这一技术得到了广泛应用，大量的商业、教育、政府、医疗以及军队组织都在使用它。德尔斐技术有很多变式，但大多由以下步骤构成：首先，征募专家，把问题呈现给他们，通常通过信件或者计算机网络来实现。其次，每名专家提出他认为最合适的解决方案。第三，组织者汇编并复制所有专家的建议，把这些建议返回或发送给专家，在专家间实现共享。第四，每名专家评论他人的观点，并提出改进的解决方案。这些改进的解决方案再次返回给组织者，组织者再次对其进行汇编并反馈给每名专家。这样的方法重复多次，直到每名专家的意见都不再发生改变，表明他们都将坚持自己的观点。采用德尔斐技术的好处是，可以收集世界各地专家的意见，扩大了信息的来源和范围，还节省了召集专家面对面交流的成本和困难。德尔斐技术已经被广泛应用于建筑、医药等企业中，帮助开发有市场潜力的产品和服务，预测技术的发展趋势。它也可以被应用于政治、经济、社会和文化领域，解决大量决策方面的问题。

名义群体技术和德尔斐技术的主要差异是，后者的成员往往互不相识，也没有直接接触的机会；而且，前者可以在几个小时内完成，而后者需要更长的时间。与面对面交流的互动群体相比，名义群体技术和德尔斐技术可以减少从众压力，避免地位高的领导支配整个讨论。研究表明，名义群体技术和德尔斐技术产生的想法数量多于传统的群体决策技术，而且名义群体技术的预测准确性与许多专家预测的准确性很接近。但是，也有研究发现，在互动的群体成员都清楚面临的任务，且不存在支配者阻碍观点交流的情况下，名义群体技术没有面对面互动的效果好。面对面交流的好处在于，群体成员对决策结果的接受程度更高，更愿意去执行它。总的来说，尽管名义群体技术和德尔斐技术并非无懈可击，但它们解决了互动群体中的群体盲思和个人支配等问题，因此不失为做出明智决策的有效方法。

四、改进组织中的决策

阻碍组织做出有效决策的因素之一是个体的偏差。如果决策者能够认识到这些偏差的

存在，并避免受到它们的影响，组织决策的效率就可以得到提高。除了前面已经提出的改进个体决策的一些建议外，还可以通过对群体成员的训练来改进组织决策。研究表明，当训练组织成员避免了以下四种常见的错误时，群体在解决创造性问题时的错误大大减少了：（1）过分追求迅速找到解决方案；（2）执着于第一个想到的方案，不去深刻地评估后果；（3）迅速改变主意，采取出现的第一个新观点；（4）回避从事手头的任务。这类训练方法可以提高个体的决策技能，从而改进组织决策。

干扰组织决策的第二类因素是群体决策的一些缺陷，主要是群体极化和群体盲思。管理者需要清醒地认识到这两种现象的潜在危害，并力求避免受它们的不利影响。

组织本身的一些限制也会影响决策。这些限制包括：（1）时间限制：许多重要的组织决策需要在时间紧迫的条件下完成。这种情况下，决策者试图迅速做出决定，因而限制了对信息的搜索和对各种可能性的考虑。尽管时间限制常常是组织必须面对的真实情况，并非是人为设定的，但它确实在一定程度上会影响决策效果；（2）“保全面子”（face-saving）的压力：决策者可能做出使他们显得面子上好看的决策，但这种决策不一定对组织的发展有利。例如，如果正确的决策将危及他的声誉或威信，一名雇员可能会歪曲决策所需的信息。一项研究发现，在一项群体决策问题上，一群商业人士选择了一个足够好的（但不是最理想的）方案，而不是做另一种选择，因为那种选择可能造成与同事的严重冲突。显然，人们经常做出一些对组织并非最佳的决策，目的是培养他们自己良好的形象；（3）管理者的决策会受到绩效评估标准和奖励体系的影响。例如，如果学校规定学生不及格的人数超过 10%，就要扣发教师的部分奖金，那么多数教师会决定不让太多学生不及格。如果组织奖励的是安全和稳健的行为，那么管理者更可能会做出保守的决策；（4）过去做出的决策通常会作为前提条件，限制当前的决策；（5）组织的正式规则。多数组织都设立各种规章制度、操作程序和其他规范，以保证个体取得较高的业绩水平，但同时也限制了决策者的选择权。作为管理者，应当意识到这些因素的存在，尽量减少可能危害决策的组织限制。

最后，一些计算机化的决策工具也可用于帮助管理者改进决策。这些工具有的是帮助个体决策，有的帮助群体决策。专家系统（expert systems, ES）是一种程序化的决策工具，它通过模拟专家的知识系统和决策过程为决策提供帮助。例如，在一个为帮助医疗诊断而设计的专家系统中，包含了由各种症状及诊断标准构成的数据库和一系列决策规则。这个系统是交互式的，用户可以通过终端机或个人电脑进入系统寻求帮助。决策支持系统（decision support systems, DSS）可以帮助管理者做出复杂的决策。它是一套计算机交流系统，加工输入的数据，并整合相关的信息供管理者使用。DSS 并非将决策转为自动化过程，而是为决策过程提供支持。它是一个动态的系统，随着决策者的使用，系统会发生变化。它也可以针对决策者制定决策的方式，量身定做。根据不同的决策者和不同的决策类型，一个组织可以拥有多个 DSS。这一系统已经被用于销售预测、货运飞机航班时间表安排和医疗决策等领域。群体决策支持系统（group decision support systems, GDSS）则是使用计算机支持群体决策的工具。另外，现在很多管理者是在虚拟的工作场所工作，不受地理、时间和组织界限的限制，通过通信和信息技术的结合来完成任务。虚拟群体的成员很少碰面，也不固定。虚拟群体通常使用桌面视频会议系统（desktop video conferencing systems, DVCS）、GDSS 和互联网/内联网系统来进行决策。尽管这些计算机辅助技术的效果仍有待

证实，但它们为提高组织决策的质量提供了新的可能。

本章小结：

1．决策是指从多个备选方案中做出选择的过程，它包括问题识别、定义目标、进行预决策、开发备选方案、评估备选方案、做出选择、执行选择和追踪调查等八个阶段。

2．决策具有一定的个体差异和文化差异。根据个体的思维方式和对模糊的忍耐性，可以划分出四种主要的决策风格，分别是指导型、分析型、概念型和行为型。不同文化背景下人们对时间的看法和使用的决策主体有所不同。

3．根据决策的常规性，组织决策可以分为程序化决策与非程序化决策；根据包含的风险，可以分为确定决策和不确定决策；根据组织中做出决策的主体，可以分为组织管理严密的决策和充分授权的决策。

4．由传统经济学的理性假设派生而来的标准决策模型强调人们的理性和信息的确定性，追求最优化的结果。西蒙的有限理性假设认为，人们的理性资源不足以保证做出最优化的决策，因而应该追求满意的而非最优的结果。卡尼曼和特韦斯基提出的前景理论更准确地描述了人们的实际决策行为。

5．人们的创造性存在个体差异，组织中的激励可以提高创造性决策。直觉决策是自动的，在某些情况下，直觉决策可以提高决策水平。

6．个体决策存在很多可以预测的偏差，包括证真性偏差、过度自信、易得性偏差、代表性偏差、锚定偏差、事后通偏差、随机性错误、承诺的逐渐升级、框架效应等。改进个体决策的方法有：分析情境、认识偏差的存在、结合理性分析与直觉、采用与工作要求匹配的决策风格、提高创造性。减少决策偏差的方法有：专注于目标、寻找反驳自己信念的信息、避免从随机事件中寻找含义、增加备选项。

8．群体决策有许多规则。群体决策既有优点又有缺点，是否采用群体决策取决于具体的决策情境和衡量决策效果的标准。

9．危害群体决策的两种常见现象是群体极化和群体盲思。群体讨论会加强群体成员的初始倾向，造成更极端的观点。当群体凝聚力非常高，且相对不受外界影响时，很容易出现群体盲思。贾尼斯探讨了群体盲思的可能原因、主要症状，并提出了避免群体盲思的建议。

10．适当使用头脑风暴、名义群体和德尔斐等技术，可以提高群体决策的质量。

11．改进组织决策的方法有：改进个体决策、训练群体成员、避免群体极化和群体盲思的不利影响、去除组织本身的限制因素、使用计算机决策支持工具等。

本章思考题：

1．分析决策的概念，举例说明决策的一般步骤。

2．举例说明两种决策模型的异同及优缺点。

3．简述前景理论的内容及其启示。

4．简述个体决策偏差，举一例说明实际工作中如何避免这些偏差。

5．比较群体决策与个体决策，举例分析各自适合的情境及其优缺点。

6．群体决策常见的问题有哪些？如何避免这些问题？

7．举例说明改善组织决策的一般方法。

案例分析：

蓝色巨人掀起头脑风暴

—IBM 公司召集 10 万员工在网上畅想创新（资料来源：Hempel, 2006）

集体智慧的威力取决于集体规模的大小，IBM 的员工就是世上最为精明强干的一个群体。熟谙此道的公司首席执行官塞缪尔·帕米萨诺打算充分发挥这一群体的作用。他的做法是与该公司近 10 万名员工全力以赴合办一个互联网会议，借助他们的聪明才智推动一场声势浩大的技术创新运动，变革行业格局，改变人们的行为，并为 IBM 带来新型的商业经营模式。帕米萨诺将公司的这项计划称为“创新工作研讨会”。

诸如此类的活动对 IBM 来说可不是什么新生事物。早在 2001 年，该公司就曾举行过网上头脑风暴会议，专门讨论如何寻找新商机，2002 年和 2003 年，又分别讨论了提升管理工作效率以及 IBM 价值体系的问题。然而，本届会议的规模可谓空前，而且首次采取公开征求意见的形式。为了发掘新的理念和构想，IBM 甚至邀请客户、行业咨询人员和员工家属共同参与有关公司未来信息技术发展的研讨。这次会议分成两个阶段进行，分别持续 72 小时，会议期间提出的任何奇思妙想都不归 IBM 所有，通过网络建言献策的任何个人都面对着一个公平竞争的氛围。

帕米萨诺披露，IBM 将出资 1 亿美元奖励提出最佳建议的人。他告诉公司员工们说，这是有史以来“第一家技术公司对外公开它所最为珍惜的商业秘密，并且告诉世人它要如何行事”。

对于 IBM 来说，大规模的经营创新和技术突破从没有像现在这样重要。该公司在持续创新方面名不见经传，因而未能吸引外来的投资。公司最近 3 年来的股票业绩亦毫无起色，7 月 24 日，即这次网络会议开幕当天，收盘价格只有每股 75.99 美元，近期的收入增长主要得益于公司压缩成本的努力，而并非开展新型业务的结果。随着 InfoSys 技术公司等亚洲企业开始蚕食其核心业务领域，IBM 急需把新型产品和服务推向市场。

IBM 做了充分准备以确保各位受邀嘉宾上网参会。它设置了一个交互式网站，带有音响点击、网络路径指南和背景信息视频摘要等功能。今年 7 月 24 日上午 10 点刚过，这个网络大会即告开始。两位主持人联袂主持了 4 个议题的讨论，包括交通运输、医疗保健、环境保护以及金融和商业运营。最初，由于成千上万的参会者同时上网，登录网站的速度一度十分缓慢，但随后 IBM 的员工设法把上网扯闲篇的人排除在外，从而使网速大为加快。参会者就所列议题进行了广泛的讨论：有人提出用带有无线射频识别功能的手环来改进旅行者在机场待机的体验，有人主张对流行疾病设置早期预警制度，还有人建议电影院在放

映影片时应该提供诸如移动座椅和喷洒水汽等感官经历，由此又有人提问说："电影院里喷洒水蒸汽是否会损坏我的助听器？"

IBM 公开征求创新理念的方式自然也隐含着一定的风险。在该公司研讨未来发展战略之际，它的 67 个客户都对此予以高度的关注，这其中包括美洲银行、Circuit City 公司以及麻省理工学院。其竞争对手也可能会潜入这次网络会议的在线聊天室，留意别人提出的任何真知灼见。不过，IBM 市场调研部门主管信息联通的副总裁、本次会议的主要策划之一艾德 · 贝文却不这样看，他说："要是没有风险，也就不会有任何创新的可能。"

这次头脑风暴的收获可能还不仅仅是一些新想法。它将为 IBM 树立一个具有超前意识的跨国公司的形象，还将向国外竞争对手表明，在规模高达 6500 亿美元的信息技术服务市场中，IBM 依然是佼佼者。

任何头脑风暴会议都会产生许许多多的新理念，但真正能够付诸实践的理念仅为沧海一粟。IBM 的这次研讨会也不例外。在接下来的一个月里，公司将组织专人对张榜公布的各类想法进行筛选，并从中优选最有可能实施的。9 月 12 至 15 日将是这次会议的第二阶段，与会者将应邀重返会议网站，依据商业价值的标准评定其中的最佳理念。让我们对此保持关注吧。

问题：

1．IBM 为何要采用网上头脑风暴来产生创新想法？

2．在头脑风暴中，什么样的激励机制会促进新想法的产生？

3．哪些因素可能会影响此次头脑风暴的效果？

小测验：

决策风格量表（来源：Rowe & others, 1984）

对每个问题，选择你认为最好地描述了典型工作情境中的你的选项。

（1）从事工作时，我通常寻求：

a. 实际结果

b. 问题的最好解决方案

c. 新的想法或方法

d. 愉快的工作条件

（2）当面临问题时，我通常：

a. 使用过去有效的方法

b. 认真分析它

c. 试图找到创造性的方法

d. 依靠感觉

（3）制定计划时，我通常强调：

a. 我当前面临的问题

b. 达到目的

c. 未来的目标

d. 发展我的职业

（4）我通常喜欢使用的信息是:

a. 特定的事实

b. 完全和准确的数据

c. 覆盖很多选项广泛信息

d. 有限和容易理解的数据

（5）当我不确定怎么做时，我:

a. 依靠直觉

b. 寻求事实

c. 试图寻求折衷

d. 等待，后来再做决定

（6）我合作最好的人是:

a. 野心勃勃和充满活力的

b. 自信的

c. 思想开明的

d. 值得信赖和礼貌的

（7）我做的决策通常:

a. 直接而且现实

b. 抽象或系统

c. 广泛并且灵活

d. 考虑他人的需要

评分:

（1）每选一个 a，就在你的指导型类别分数上加一分。

（2）每选一个 b，就在你的分析型类别分数上加一分。

（3）每选一个 c，就在你的概念型类别分数上加一分。

（4）每选一个 d，就在你的行为型类别分数上加一分。



第四篇 情绪管理

第十章　情绪劳动与管理

20世纪中叶以前，由于生产力发展水平所限，劳动往往仅被看作一种体力的付出过程，而情绪这一影响人们生活和工作的重要心理因素并没有得到足够的重视。到了20世纪70年代，随着情绪在心理学中得到广泛而深入的研究，理论工作者和实际工作者才把研究的兴趣逐渐转移到组织中的情绪问题上来。尤其近几年，随着服务业在整个国民经济中的比重不断提高，越来越多的企业要求员工在与服务对象的接触中，适当掩藏和控制内心的真实情感，即使精疲力竭、情绪低落也必须按企业的要求微笑地面对顾客。这样，员工在劳动或工作中的情绪问题，如倦怠（burnout）、情绪失调（emotional disorder）、工作压力（work stress）和工作引起的不满（work dissatisfaction）等问题就被人为地压制起来，集腋成裘，最终威胁到员工的身心健康。在这种情况下，一个新的概念——情绪劳动（emotional labor or work）吸引了研究者们的注意。至此，劳动的内涵被注入了新的内容：劳动不仅仅是一种体力、脑力的付出，更是一种情绪资源的付出。这种不可见的资源付出作为员工工作内容不可分割的组成部分，应被纳入薪酬计量中加以计算。本章对情绪劳动的相关内容进行阐述，并对由此引发的员工心理健康问题进行了探讨。

第一节　情绪劳动概述

一、何谓情绪劳动

情绪劳动的概念最早由霍奇德（Hochschild）于1979年提出，并在其1983年出版的著作中以空姐的工作为例对情绪劳动进行了详细的分析。他认为，情绪劳动多存在于情感密集型行业（如护士、教师、服务员等），是指劳动者为了获得一定报酬而对自己的情绪进行控制，以营造出公众可以观察并接收的面部和身体上的动作表现。从这个概念可以看出，首先情绪劳动是为了获得报酬，其目的性很明确。当然，这种报酬除了金钱与物质奖励外，也可能包括晋升机会以及良好的人际关系等。其次，情绪劳动并不一定通过面部表情来实现，也可以通过身体行为及语言来表现，如规范的礼貌用语和示范性动作等。再次，人的自然情绪有多种，如哭、笑、悲伤、愉快、紧张、焦虑等。情绪劳动强调的是对自己情绪的控制，也就是控制自然情绪，使其适合特殊岗位和工作环境的要求，因而是一种特殊形式的劳动，需要当事人付出努力以实现非自然的情绪表达。

随后，又有许多学者对情绪劳动的内涵进行界定。莫里斯和费德曼（Morris & Feldman,1996）认为，情绪劳动是指人际交往中为了表达出组织所期望的情绪，员工必须

进行的努力、计划和控制等活动。第芬多夫和格罗斯朗德（Diefendorff & Gosserand,2003）则更直接地将情绪劳动定义为：为了响应组织有关情绪表现规则以完成组织工作任务而对个人情绪表现进行管理的过程。由以上论述可以看出，情绪劳动应具备以下几个条件：（1）情绪劳动需在与顾客面对面、声音对声音的互动中完成；（2）情绪表达要用来影响他人的情绪、态度和行为；（3）情绪的表现要遵循一定的规则。因此，本书将情绪劳动定义为：当组织中的员工与其他人进行面对面或语音（如电话或邮件）沟通和交往时，为完成组织任务，根据组织有关规则和要求做出特定情绪表现的过程。

二、情绪劳动的结构与分类

既然将情绪的控制视作一种劳动，那么对这种劳动的衡量就很重要。找到情绪劳动的结构是对其进行衡量的第一步。莫里斯和费德曼将情绪劳动划分为四个维度：（1）情绪表达频率，即在工作中要求表现情绪的次数。太频繁的情绪表达要求会使员工感到疲惫。（2）重视情绪表达规则。对情绪表达规则越重视，员工就会越努力地完成情绪劳动。（3）情绪的多样性，即在情绪劳动中表现出不同的情绪，如喜怒、爱憎、冷漠或同情等。一般认为，要求表现较多情绪的劳动，其情绪劳动强度就比较高。不同职业对情绪多样性的要求是不同的，一些职业（如幼儿教师、护士）要求比较丰富的情绪表现，如教师通常要表现出和蔼友善，而面对学生的错误行为时应该表现不满甚至气愤。有些职业对情绪的要求则比较单一，如法官必须表现出一贯的威严、独立和公正。情绪的多样性也包括表达同一情绪的不同强度，如平静、愉悦、兴奋就属于“快乐”情绪的三个不同强度，把握好“度”并按照要求准确表达需要付出较多的情绪劳动。（4）情绪失调。对这一维度，不同的研究者有不同的看法，有的研究者把它看成是情绪劳动的维度，有的把它看成是情绪劳动的后果，还有的把它看成是一定社会环境中的张力。如果员工被要求表现他们实际上无法真正体验到的情绪，就有可能产生情绪失调。情绪失调也可被看作是一种个人角色冲突。情绪失调是情绪劳动的负面结果。如果长期不能体验到应该体验到的心情，就有可能使虚伪成为一种习惯，长此以往会导致个人角色与自己的真实情感相背离，从而产生自卑和压抑。有研究者认为，可以从情绪失调程度和情绪努力的程度（即努力改变自己的情绪使之与组织期望的情绪相匹配）两个方面来对情绪劳动进行衡量。

根据情绪劳动所动用的情绪资源层次的不同，情绪劳动可以划分为三种类型：自发的情感行为、表层的情感行为及深层次的情感行为。以医生职业为例，当医生看到一个被病痛折磨的病人，他可能首先想到的不是“这是我的工作，我应该去帮助她”，而是本能地想去帮助她。尽管最后表现出来的都是对病人的治疗，但这种本能的行为更多地可能是出于一种“不由自主”的条件反射。这种行为被认为是自发的情感行为，是情绪劳动者在外界相关因素刺激下的一种本能的反应。但是，如果医生每天接触很多这种经受痛苦与死亡折磨的病人和家属，且日复一日、天天如此，他们会有怎样的反应呢？这就涉及到情绪劳动的频率问题。在情绪劳动频率很高的情况下，如果总是要求员工自然地投入其中，由于情绪资源的有限性，员工很可能出现情感的“透支”。所以，在高度的情感密集型环境下，太

多的外界刺激降低了员工对情感的敏感性，他们可能会逐渐变得麻木起来。但为了完成自己的工作，尽到职责，情绪劳动者也不能完全对服务对象熟视无睹或麻木不仁，他们需要通过面部表情、语言、声音及姿态等可以感知的方式来表现其承担的工作应付出的情感性劳动。与前述那种本能的自发的情感行为不同的是，这时表现出来的情感并不是其内心真正的情感，也就是说这种情感只是一种表面现象，甚至可能是一种伪装。这种类型的情绪劳动被称为表层的情感行为，也有研究者将这种表面的情感行为称为职业情感。职业情感可以使情绪劳动者在面对情绪劳动频率很高、强度很大的工作时避免发生自身情感的透支甚至耗竭，同时又满足了企业对自己的工作要求。然而，这种职业情感又会面临这样的问题：如果一些被服务者发现情绪劳动者对他们的服务仅是一种职业的伪装，那么他们的满意度就会降低，并最终影响他们对服务质量的评价。因此，要在自发的情感和职业情感之间取得平衡，既不过分耗费自己的情感资源，又不因缺乏真心而降低被服务者的满意程度，需要把握好工作情感上“真”与“假”的分寸。为解决这一问题，研究者提出了深层次的情感行为概念，即情绪劳动者在工作中有意识地管理自己的情感，一方面意识到自己的角色，明白自己正在工作，从而做出恰当的情感表达；另一方面，要适当调整自己内心真实的情感和情绪，将一些强度很高的情感转化为强度较小的情感，并适当控制情感的持续时间，在不影响对服务质量评价的情况下，减少自身情感的损耗。

与情绪劳动紧密相连的一个重要因素就是组织的规则。一般说来，每个组织对员工情绪劳动的作用都有这样的共识：员工的积极情绪有利于组织发展，消极情绪有害于组织发展。因此，提倡积极情绪，遏制消极情绪成为组织情绪表现规则的主要内容。

组织情绪表现规则包含两个维度：情绪的表现性维度和情绪的自主性维度，这两个维度都是由不同程度内容组成的连续体。如图 10.1 所示，表现性维度的一端是显性的情绪表现规则，指组织内部明文规定工作中期待员工表现的情绪，如公司倡导微笑式服务、家庭式服务等；另一端是隐性情绪表现规则，指组织虽然期待员工在工作中表现出某种情绪，但并没有明文规定，而是希望员工能通过感同身受、潜移默化地表现在自己的行为中。如组织希望员工见到领导时有恭顺殷勤的表现，这一点不可能列在组织的规章制度中，但却是组织希望的行为。自主性维度的一端是强制性情绪表现规则，是指组织强制性地要求员工工作中表现出的某种情绪；另一端是自主性情绪表现规则，指员工工作过程中自然流露出的某种情绪。这两个维度相互结合就构成了员工在面对情绪规则时的不同应对状态，如图 10.2 所示。

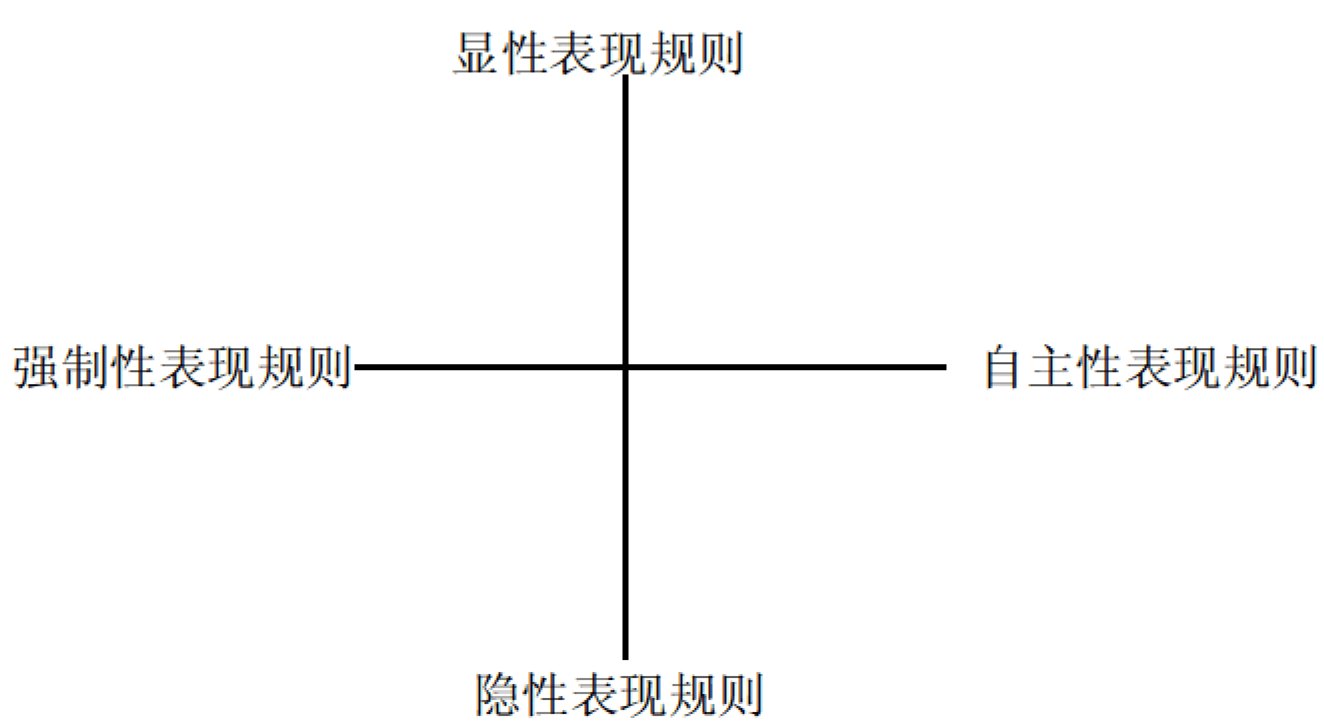


图 10.1　组织情绪表现规则

	显性表现规则	隐性表现规则
自主性表现规则	认同	内化
强制性表现	顺从	应激

图 10.2　对组织不同表现规则的应对状态

三、情绪劳动的主要作用与影响

情绪劳动受到如此关注，那么如何评估情绪劳动的价值？企业关注情绪劳动背后的原因何在？情绪劳动对企业、顾客、自身会产生什么影响呢？

（1）情绪劳动对顾客的影响。情绪劳动最核心的作用在于情绪劳动者可通过自己表现出来的情感，影响顾客的态度、情绪、情感及行为。这一点在服务性行业中尤为常见。通过服务人员的工作，如果顾客的态度、情感与行为达到了企业的预期（顾客购买了产品，企业获得了利润），那么可以认为服务人员的情绪劳动是有效的。对于顾客而言，消费过程满意与否不仅仅取决于产品和服务的功能，还取决于整个消费过程中的心理体验。这一方面来自于适当的产品或服务设计、恰当的消费环境，另一方面来自于服务人员的情绪劳动。服务人员恰当的言谈举止、真实的情感投入丰富了顾客的消费体验，满足了顾客的情感需求，使他们产生了积极的情感体验，创造了一份美好的回忆。

（2）情绪劳动对企业的影响。随着信息技术的进步和产品的日益丰富，企业仅仅通过产品的功能性服务已经很难完全满足顾客的需求，顾客越来越重视特定消费经历对内心情感和思想的触动。新加坡航空公司在此方面有相当成功的经验，公司十分注重选拔、培养和留用周到细致的空中小姐，她们美丽的笑容和对顾客周到细致的关怀和服务，使新加坡航空公司创造出"一种全新的飞行体验"，为此公司对她们支付高薪，并予以提拔重用。可见，为顾客创造美好的情感体验是企业增强自身竞争优势的途径之一。这就需要企业做好服务人员的选拔、任用及培养等工作，并通过建立情感性的企业文化和相应的制度来激励服务人员的情绪劳动，增强顾客的消费体验，从而提高顾客的满意感和忠诚度。

（3）情绪劳动对服务人员自身的影响。一些研究者分析了服务人员的情绪劳动与其情感消耗之间的关系，结果发现，医生、教师、服务员等行业的劳动者为了做好份内的工作，在面对服务对象时，常常需要大量的感情投入。这种投入很容易使他们过度消耗自身的情感资源，表现出抱怨、精神压抑甚至人格解体等现象，从而降低其工作满意度，甚至产生心理问题（情绪劳动与心理健康的关系将在下一节详细探讨）。但是，如果这些付出可以得到企业的奖励，如更多的报酬和精神上的鼓励等，或者如果服务对象做出了积极的反馈，甚至彼此之间建立起良好的个人关系，那么服务人员的工作满意感就会相应提高。因此，建议企业将员工付出的情绪资源纳入薪酬计量之中，同时给予他们更多精神上的关怀，使情绪作为一种可再生资源不断投入到工作中去。

第二节　情绪劳动与员工心理健康

长期以来，企业关注得较多的是员工的身体健康，而对心理健康问题有所忽视，对职业心理保健投入很少。但是近年来，随着全球化竞争时代的到来，人们的工作、生活节奏越来越快，数字时代大量信息的冲击、人口爆炸带来的严峻就业形势、生态环境的恶化、林林总总的诱惑和选择、对职业发展和企业前景的疑惑、未来社会政治经济发展预期的不确定性，都使劳动者的心理健康问题凸显出来。心理亚健康、不健康的状况越来越困扰着企业员工，压抑、抑郁、焦虑、烦躁、苦闷、不满、失眠、恐惧、无助、痛苦等不良的心理反应像幽灵一样时时困扰着上至管理层、下至普通员工的心，严重时它可以使得企业停滞不前，在严重的“内耗”中走向灭亡。在这种情势下，员工心理健康问题引起了越来越多企业管理者的重视，员工的心理保健工作已经迫在眉睫。

一、情绪劳动中的心理健康问题

情绪劳动者时刻进行着情感强化和情感置换的过程：一方面要增强自己和服务对象之间的亲密感，把陌生的服务对象想象成自己的朋友和亲人，对待他们像对待自己的亲人一样；另一方面则要隐藏起自己的真实情感，正确表达企业需要的情感。即使员工不喜欢甚至讨厌某个人，但一旦这个人成为自己的服务对象，仍然要流露出高兴快乐的表情。在这种情况下，员工长时间压抑自己的真情实感，很可能出现一些不良的后果，主要表现在情感耗竭、去人格化和工作满意度降低三个方面。

情感耗竭是指情绪劳动者由于角色负担过重、角色冲突强烈等问题而心情烦闷，甚至长期精神压抑，感觉自己已经被“掏空”，无法再继续付出的状态。这在医护人员的情绪劳动中出现的频率相当高。去人格化是指在需要投入的情绪劳动频率过高、持续时间较长的服务中，情绪劳动者容易将服务对象当作一个需要服务的物体而不是活生生的人来看待。此时情绪劳动者的情感表达也因此而成为一种程序化的无意识反应，他们表面上所表现出来的情感反应与内心的实际感受完全分离，他们的微笑和热情都只是为了完成份内工作而做出的一种机械动作。过度的情绪劳动还会降低服务人员的工作满意感，表现在对工作没有劲头，提不起精神，离职倾向明显等。

除此以外，社会学家和心理学家在研究了快餐店、航空公司、银行、酒店、医院等行业从事服务工作的员工后，发现情绪劳动者在进行角色转换过程中还存在以下问题：

（1）角色混淆。长期从事情绪劳动的员工要按照企业要求，掩藏个人真实情绪进行角色转换，他们可能分不清楚哪些情感是属于自己的情感，哪些情感是与工作相关的情感。如餐馆服务人员在与顾客交往过程中，始终保持友善的微笑、愉快的心情，活跃就餐气氛，镇定地应付突发事件和尴尬局面。在他们下班以后，他们可能仍然分不清哪一份情感是与工作相关的，哪一份感情是他们自己真实的情感。他们甚至说不清自己的真实感受，或者

总感觉自己的情感不是特别的真实。长此以往，员工就会感到工作压力或对工作产生厌倦。

（2）角色冲突。服务性行业常常有许多明文规定的制度和情感服务条例用来规范员工的言行举止，员工为了更好地履行这些规则往往会将它内化为个人的行为准则。在员工下班后，如果不能及时转换角色，就会导致工作状态下和生活状态下的角色冲突。银行出纳的工作繁忙而紧张，排队等候的顾客经常对他们言语粗暴，因此银行出纳在 8 小时工作中始终处于极度消极的情感状态。下班后，如果他们能及时调整工作上压抑紧张的情绪，在轻松祥和的家庭氛围中保持积极的心态，就可以在相当程度上减少对心理健康的损害；如果不能，就会把消极情绪带回家，影响家庭的氛围和和睦。

（3）角色分离。为了给顾客提供个性化的服务，企业往往要求员工与顾客之间建立一种“虚拟关系”，增进相互之间的亲密程度，建立顾客对企业的信任感与归宿感。然而，长时间高强度的情绪劳动可能导致与企业期望相反的效果——员工的工作角色与其自身的感受出现分离与脱节，员工不能表现自己的真实情感，使得顾客体会到的只是一种职业化的服务，而不是一种发自内心的关怀。

二、情绪劳动中的自我管理策略

在情绪劳动中，员工如何进行情绪管理以降低对身心健康的损害呢？研究者提出了四种策略。

（1）自主调节。这种策略把情绪看作一种自主的体验过程，当事人没有必要去刻意扮演，以自动模式来完成情绪表达即可。在感觉到某种情绪时就自然流露，产生相应的情绪反应，这种情绪劳动不需要有意识的努力。如幼儿园阿姨看到小朋友哭泣，就会很自然地进行安慰，这种行为几乎成了一种下意识的表现，不需要主观努力。

（2）表面扮演。大多数情绪研究者都承认情绪是由主观感受、心理反应和表情行为等几个部分构成的，表情行为包括面部表情、言语表情和姿势表情。表面扮演就是员工尽量调控表情行为以表现组织所要求的情绪，而内心的感受并不发生改变。表面扮演意味着内心的感受与外部表情之间的分离，但不失为一种策略。因为在无法改变现有的内心感受时，就必须以表面的装扮来开展工作，这就是我们常说的“陪笑脸”。对于一个表面扮演水平较高的员工来说，掩饰某种情绪是一件很容易的事，不会体会到太多的痛苦。但是，如果扮演拙劣的话，就可能被对方识破。

（3）深层扮演。当表面扮演因为太机械而不能满足顾客对真诚人际关系的需要时，深层扮演就变得非常必要。深层扮演策略就是为了按要求进入角色，尽量去体验必须产生的情绪。在这种情况下，表情行为是发自内心的。此时，不仅要对表情行为，而且要对内心感受进行管理。深层扮演是一个积极主动的过程，要求员工尽可能努力激活那些能够引起某种情绪的思想、想象和记忆等心理活动，精心调控情绪，做到表里如一。例如，一名售货员在家与丈夫吵了架，心情抑郁、气愤，但在即将走上工作岗位面对顾客时，她就必须努力排除抑郁和气愤的心情，想象或回忆可以诱发积极情绪体验的情境，如结婚时的热闹

场面、在海边度假时的心情等，在内心真正体验到快乐时，就能面带笑容地面对顾客。

（4）失调扮演。这种策略要求员工保持平静的中性心情去应对各种环境刺激，以便集中精力去完成主要任务。它要求员工能够表面上表现出职业所需的情绪，而内心仍保持中性。因此，员工必须对自己的情绪进行管理，不让坏情绪的刺激和表面的情绪影响自己的心情。内心的体验与外部的情绪表达不一致，这种内外失调反而更有利于工作（完成主要任务），这就是常说的“职业性公关”。例如，一名银行经理在为一个几百万元的贷款项目进行谈判时，尽管在谈判过程中有各种各样的表情，但内心却一直保持镇静和中性，任尔东西南北风，我自岿然不动。

第三节　员工工作倦怠分析

在过去的二三十年间，工作倦怠（job burnout）已成为西方组织行为与人力资源管理领域研究的热点问题。特别是近几年来，工作倦怠已成为美国企业人力资源会议上讨论的主要议题之一，引发了众多学者对其理论与模型进行广泛深入的讨论。

一、工作倦怠的概念及结构

对工作倦怠的研究始于 20 世纪 70 年代。1974 年，弗罗登伯格（Freudenberger）首先提出了倦怠（burnout）一词。他认为，倦怠是一种情绪性耗竭的症状，这种症状最容易在工作情境中出现。当工作本身对个人的能力、精力以及资源过度要求，从而导致工作者感到情绪枯竭、筋疲力尽时，工作倦怠就产生了。

在弗罗登伯格的概念基础上，许多研究者提出了自己的观点。马斯拉奇（Maslach）在对专业助人行业（如教师、护士、社会工作者等）进行研究后，提出了工作倦怠的三维度理论模型。该模型认为，对于从事这类服务性职业的人，人们往往存在一种期望，希望他们在工作中一贯地为他人提供优质的服务。这种社会期望造成他们在进入职业之后不得不持续地投入大量的情绪、生理等方面的精力来应付服务对象的要求，久而久之，就容易形成身心俱疲的症状。马斯拉奇把这种由于长期对情绪及人际应激源做出反应而产生的心理综合症称为工作倦怠。在对服务行业进行访谈及个案研究的基础上，1981 年马斯拉奇编制了工作倦怠量表（MBI），并将工作倦怠从三个维度加以定义，这三个维度分别是情感耗竭（emotional exhaustion）、去人格化（depersonalization）、个人成就感降低（diminished personal accomplish）。其中，情感耗竭是这一系列症状的主要方面，被认为是倦怠最具代表性的指标。它的特征是缺乏活力，有一种情绪资源耗尽的感觉。此外，情感耗竭经常伴随着挫折、紧张，所以员工会在心理层面上自认为无法致力于工作。去人格化是工作倦怠的第二个维度，其特征是视服务对象为“物”，而不将其当成“人”看待，表现为对他人消极、冷淡、过分隔离、愤世嫉俗等态度和情绪。去人格化是员工过度紧张或耗竭时的一种防御性反应。

第三个维度是个人成就感降低，表现为自我效能感降低，自尊心下降，感觉无助，倾向于对自己做负面评价，工作满意度也随之降低。

马斯拉奇的研究在一定程度上揭示了工作倦怠的结构特点，但由于其研究对象仅限于助人行业，因此其普适性受到了限制。在后来的研究中，他将 MBI 量表在其他非专业助人行业中进行检测，修订为 MBI-GS（MBI-General Survey）量表，并将原有理论中的三个维度调整为耗竭（exhaustion）、疏离（cynicism）和无效能感（inefficacy）。耗竭是指个体的心理资源（包括情绪资源和认知资源）被耗尽的感觉；疏离，特别是工作中的疏离本是组织行为学中一个独立的概念，这里被用来描述个体产生的对工作多个方面（包括对人、对事等）的一种消极的、冷漠的或与工作极度分离的反应；而无效能感则是自我效能感低、缺乏成就感与创作能力等几个方面的综合。马斯拉奇提出的新三维度理论在一定程度上反映了工作倦怠现象的本质，成为目前工作倦怠研究领域广为接受的、影响最大、居主导地位的理论模型。

还有一些研究者对工作倦怠的定义和结构进行了探讨。平斯（Pines）等人在着手研究之初，就将倦怠定位于生活的方方面面（如婚姻关系、政治冲突等），而不是仅限于专业助人行业。从这个意义出发，他们把倦怠定义为：由于个体长时期处在对其情绪资源过度要求的情境之下而产生的一种生理耗竭（physical exhaustion）、情感耗竭（emotional exhaustion）和精神耗竭（mental exhaustion）状态。平斯等人认为，不论倦怠的程度如何，它所包含的生理、情感及精神耗竭这三个基本的成分不变。该理论仅关注马斯拉奇三维度结构中的耗竭维度，故也被称为单维度理论。

马斯拉奇和平斯对工作倦怠结构的划分虽然都得到了一些学者的认可，但他们都是从实证研究角度提出自己的模型的，缺乏必要的理论依据。施罗姆（Shirom）等人从应激资源守恒理论出发，对工作倦怠产生的渊源进行了理论解释。应激资源守恒理论认为，人们拥有一些有价值的资源，如客观资源（汽车、住房）、条件资源（朋友、婚姻、权力）、个人资源（自我效能、自尊）、能源（时间、金钱与知识）等，人们为了实现一定的目标，总是试图保存已有的资源并渴望获得更多的资源，当资源受到威胁或不足以应付需求、资源失去或个体感到失去，以及进行了投入却没有收获时，就会出现心理应激。应用到企业当中，当个体在长期的工作中源源不断地耗费了大量的精力资源却不能得到及时的回报和补充时，工作倦怠的状态就可能产生。这里的精力资源包含生理的、情绪的及认知的精力。因此，施罗姆等人将工作倦怠定义为：表现为身体上的疲惫、情绪上的衰竭与认知上的厌倦相结合的一种状态。这样就将工作倦怠的概念进一步明晰化，减小了它与其他变量的重叠。从本质上讲，它也是一种单维度理论。

除以上谈到的几种主要研究之外，近年来，还涌现了一些有关倦怠问题的拓展性研究。邓士登（Densten）从 MBI 量表中耗竭、疏离与无效能感维度出发，提出了三个方面的假设：（1）耗竭维度应该由心理紧张与躯体紧张两方面的因素构成；（2）疏离维度应该由对工作的疏离和对人的疏离两个方面因素构成；（3）无效能感维度包括自我评价和他人评价两个方面。经验证，最终得到了一个五因素的工作倦怠模型，包括心理紧张与躯体紧张、自我评价的无效能感与他人评价的无效能感以及疏离。此外，莫尔（Moore）等人进行了

工作倦怠归因分析、工作倦怠与工作压力之间的关系分析等研究，也为该领域的研究注入了新鲜血液。

二、工作倦怠的影响因素

关于工作倦怠形成的原因，存在两种不同的观点。第一种观点称为工作环境成因论，认为工作环境等外部因素是倦怠形成的主要原因。按照应激资源守恒理论，当工作环境等外部因素对个体的要求持续超过个体具有的有效应对资源时，就会发生资源枯竭。这些因素包括：工作角色冲突与模糊、工作负荷、工作可控性、报酬、工作群体、社会支持系统、时间压力、组织公平、组织与管理变革以及需要持续、频繁、深入地与工作对象接触等。其中，长期超负荷工作、工作环境差、报酬低于自己的期望值是导致工作满意度下降和工作倦怠形成的主要原因。经常体验到组织不公平感和对组织管理变革采取消极态度的个体更容易产生倦怠。

与工作环境成因论不同，第二种观点称为人格特质论，认为个体的人格、态度等内部因素以及某些人口统计学变量可以更有效地解释倦怠的形成。即使面对相似甚至相同的工作环境要求，不同的个体发生倦怠的情况也存在显著差异。性格被动、退缩和沉默寡言的个体容易受到情感耗竭、成就感降低的困扰；不合作、疑心重和易激惹的个体在去人格化方面表现较明显；对组织、工作以及自己的绩效表现抱有很高期望或者期望水平变化波动较大的个体发生倦怠的几率比较高。此外，低勇敢性、外控、防御性应付、低自尊、A型行为模式等人格变量与倦怠关系密切。在人口统计学变量方面，从年龄来看，刚刚开始职业生涯的年轻人比已经工作了一段时间、30岁以上的人易发生倦怠；女性比男性易发生倦怠；教育水平高的个体比教育水平低的个体易发生倦怠；未婚个体发生倦怠的可能性大于已婚者；对已婚者而言，没有子女的个体比有子女的个体更易发生倦怠。

对于以上两种观点的差异，笔者认为：工作倦怠本身是一个多维度的概念，其形成可能受到许多因素的影响；在倦怠形成的过程中，这些影响因素之间也可能存在复杂的相互作用；另外，它们在倦怠形成中的作用还可能受到其他中介变量的调控。所以，应该采用多特质、多方法、长期追踪式的研究范式，同时考察多个变量在倦怠形成中的作用，并采用严格控制条件下的实验研究才能得出确凿的因果关系的结论。

三、工作倦怠的干预措施

对于工作倦怠的干预策略，不同学者以不同理论为基础，提出了不同的看法。

从资源论角度，将干预策略分为以人为中心的干预策略和以工作情境为中心的干预策略。前者强调加强员工的内部资源，从认知角度改变个体应对压力源的反应，包括减少期望、重新解释工作意义，以及培养乐观态度、组织承诺感、正性情绪和主动应对策略（问题解决的积极导向）等。此外，采取营养支持、心理调节、运动保健等养生措施促进身心

健康，平衡工作生活休息时间，发展放松的生活方式等都可以增强应对倦怠的个人资源。以工作情境为中心的干预策略强调从减少工作负荷、角色冲突、角色模糊等工作需求和加强社会支持、组织公平、工作自主性等工作资源两个方面进行干预。

从努力—奖酬平衡论角度来看，员工投入的个人努力和获得的奖酬之间的不平衡是倦怠产生的根源，因此可以从个人努力和奖酬系统两方面对倦怠的产生进行干预。在个人努力方面，减少工作时间，提供休息室、修养娱乐设备等是很好的干预措施。在奖酬方面，不仅需保证公正的物质性奖酬，还要注重提供非物质性奖酬，其中保护员工自尊是较敏感的奖酬成分。领导通过倾听、及时反馈和提供社会支持等能让员工认识到自己工作角色的重要性，获得自尊感的满足。另外，实施员工帮助计划（EAP），包括压力管理、裁员心理危机干预、职业生涯管理等，可以减少工作不安全感，提高自尊，减轻员工压力，提高员工的工作绩效。

从匹配理论角度看，人与工作的不匹配造成了倦怠的产生，因此就应该从匹配角度实施干预措施。例如，有的员工十分注重工作意义和对努力的奖励，能容忍较大的工作负荷，相应地，干预措施就应该主要从价值和奖酬方面设计。同时，应将组织特征干预与教育干预结合起来，改变员工的消极态度和技能，实现匹配。在工作负荷匹配方面，培训员工学会合理分配工作，在员工能力、时间和精力之间找到最优的平衡点。在奖酬匹配方面，提供与员工努力相匹配的报酬，包括内在奖励和外在奖励。在社会交往匹配方面，进行人际交往培训，提供丰富的组织支持资源，满足员工社会联系、交往分享的需求，减缓倦怠感。在公平匹配方面，不仅需要公平合理的任务分配和奖酬分配制度，而且在程序公平上，赋予员工发言权，建立参与管理和合理的绩效考核体系。在价值观匹配方面，招募价值观与组织一致的员工，可以降低员工在后来工作中倦怠发生的几率；进行工作再设计和任务分配时，提高员工的价值观和任务特征匹配度。在员工与组织匹配方面，招聘新员工时进行现实工作预览（RJP）和期望降低计划（ELP），调整新员工的期望值，与接下来的工作经历相适应，不仅可以预防倦怠感的产生，而且有助于增加员工的组织承诺，降低离职率。

管理小贴士：

日本员工心理保健的新方法——产业心理咨询

产业心理咨询是员工帮助计划（EAP）在日本的变形和再应用。之所以称之为产业心理咨询，而不是企业心理咨询，主要是因为产业的概念更为广泛，包括第一产业、第二产业、以及以服务为主的第三产业，工商企业、服务机构、学校、警察、金融、医院、政府部门等都包含其中，涵盖了全社会的职业类型。

日本的产业心理咨询已经有五十多年的历史。企业一般都设有咨询室、恳谈室、宣泄室或茶室等，还有一些企业引进了心理服务，包括心理咨询、心理测试、心理援助等。松下公司著名的“出气室”就是一个典型的例子。在这个房间里有松下幸之助的仿真橡皮人，供有怨气的员工打击以发泄内心的情绪。当员工打击完橡皮人之后，那橡皮人嘴里就会响

起松下幸之助本人的声音，这是他写给员工们的一首诗。诗中说：这不是幻觉，我们心心相通，团结合作，因为我们都生在日本国。我们有一致的目标——让日本繁荣幸福。我们的做法可能有分歧，但必须记住祖先的忠告：让民族强盛、子孙和睦。松下把“出气法”的心理宣泄与爱国主义教育紧密结合起来，把与员工的沟通、理解凝聚到民族精神上，可以说是一个明智而巧妙的做法。

产业心理咨询服务的主要内容，一是员工的心理健康；二是员工的压力管理和员工职业生涯的规划与发展；三是员工工作场所的人际关系调试。产业心理咨询服务的对象是所有的劳动者和与劳动者相关的人，包括员工的亲属。通过对从业人员及其亲属的心理问题进行辅导，解决他们的问题，增进其健康，提高其生活质量。

在日本，从事产业心理咨询的人员是一批训练有素的产业心理咨询师。他们的年龄大多在 40 ~ 50 岁，专、兼职的都有，男女基本各半。这些人的出身背景有学医学的、有学护理的、有专门学产业咨询的、有学心理学的。产业心理咨询师在企业里一般身兼数职，担当着咨询师、训练者、员工福利顾问员等多重角色。他们与企业的人力资源管理部门密切合作，开展职业心理咨询、心理健康咨询、员工个人能力开发和自我发展等工作。要想成为一名合格的产业心理咨询师，必须经过严格的资格培训和资格认证。初级咨询师必须是大学心理学本科毕业，中级咨询师起点是硕士，高级咨询师是博士。如非心理学本科、硕士、博士，则必须具有很多年的工作经验并经过严格的心理咨询专业的培训和考核。

日本的产业心理咨询，一般在企业设有专门的机构，配备有一至几名专门人员。机构设置各个企业不尽相同，有的设在总务部，有的则设在人事部，有的设在健康管理部。名称叫法也不尽相同，有叫职员咨询中心的，有叫职员咨询室的。产业咨询师在工作中经常遇到的问题是身体健康、工作场所不适应、人际关系困扰、岗位和职位改变、精神病、酒精或其他药物依赖、工作压力、离婚、中年期抑郁症、退休适应等问题。

四、工作倦怠研究的新趋势

（一）研究内容上的扩展

将工作投入（job engagement）的概念补充进工作倦怠模型，把倦怠与投入看作是一个三维连续体的两个极端点，这三个维度分别是精力（energy）、卷入（involvement）和效能（efficacy）。位于其中一端的是工作投入，指一种感觉充满精力，能有效地进入到工作状态并与他人和谐相处的状态，这对个体与组织而言都是积极的、高效的工作状态。而在相反的一端则是倦怠，一种感觉无效能、耗竭以及与工作及他人疏离的状态，在这种状态下，个体的精力变得耗竭，卷入程度低，效能感丧失。将工作投入作为工作倦怠的一个积极的相反状态来加以研究，扩大了工作倦怠的研究范围，通过提升对工作的投入程度来使员工远离工作倦怠，这将是今后研究的一个重要领域。

（二）研究视角的扩展

当前的企业正处于一个不断变化的外部环境之中，传统的组织结构形式受到了严峻的

挑战，众多的新型组织形式，如团队、虚拟组织、无边界组织等纷纷涌现。组织结构的变革，必然引起组织成员心理上的种种变化，工作倦怠的影响因素及具体表现也会有所不同，这也要求组织和个体采用相应的措施来加以控制和干预。例如，自我管理型工作团队是组织成员被赋予完成整个任务的责任，并且可以自主决定如何去完成任务的新型组织形式。团队成员除了要完成技术上的任务外，还要分心出来决定与工作任务相关的其他事情，如任务的分配、执行任务的方法、活动的计划等，有时还要花费大量时间去解决与工作相关的人际事务。这就要求团队成员之间进行更为频繁与密集的互动，同时关注多个层面上的关系，特别是与其他团队成员的关系，这很可能成为工作倦怠的重要引发因素。当然，自我管理型团队也有一些因素对于减轻工作倦怠是有帮助的，如个体的任务非常清晰、个体的绩效情况能得到及时全面的反馈、成员之间能得到较多的社会支持等。总之，对于新型组织结构下的倦怠问题，需要我们针对相应组织形式的特点，进行具体的研究，这将是今后工作倦怠研究的一个重要课题。

（三）研究区域的扩展

倦怠研究区域的扩展主要表现在跨文化研究上，由原来仅由美国学者研究转变为受到各国研究者的普遍关注，跨文化的比较研究也日益得到重视，并已取得了一些成果。例如，在欧洲工人中进行的几项研究发现，在情感耗竭维度上，欧洲的样本要比北美样本的得分低一些。产生这种差异的原因，一方面可能来自于文化价值观的不同，因为公开表达自己关于倦怠方面的一些观念，在强烈个人主义的北美社会中要比更注重群体一致性的欧洲社会中更容易被他人所接受。另一方面也有可能是，成就导向更强的北美社会对他们的工作感觉到更大的压力。工作倦怠的跨文化比较研究目前还处于起步阶段，在研究内容与方法上还基本上继承美国学者所建立的概念体系及研究方法，测量上亦多采用 MBI 的各国翻译版本，因此从学术意义上看对倦怠的探讨还不够深入，适合各国国情的倦怠的替代性测量工具还很少开发出来。研究中样本的选择往往缺乏代表性，所发现的跨文化差异及其相应原因的推断尚相对缺乏说服力。

本章小结：

1．情绪劳动是指当组织中的员工与其他人进行面对面或语音（如电话或邮件）沟通和交往时，为完成组织任务，根据组织有关规则和要求做出特定的情绪表现的过程。

2．情绪劳动可以划分为三种类型：自发的情感行为、表层的情感行为及深层次的情感行为。

3．情绪劳动对员工心理健康的影响主要表现在情感耗竭、去人格化和工作满意度降低三个方面。

4．马斯拉奇提出了工作倦怠的三维度结构模型，并引发了后人一系列的研究。

5．工作倦怠研究出现了研究内容上的扩展、研究视角的扩展、研究区域上的扩展三方面的新趋势。

本章思考题：

1．简述情绪劳动的概念。为什么情绪劳动会引起人们的关注？
2．如何对情绪劳动进行分类与衡量？
3．简述情绪劳动的作用和意义。举例说明如何管理情绪劳动。
3．简述工作倦怠的概念。
4．试述关于工作倦怠结构维度观点的演变过程。
5．试述工作倦怠的影响因素。
6．结合管理实践谈一谈如何缓解员工的工作倦怠。
7. 根据你自己的体会或你在商业杂志中读到的材料，描述现实世界中工作倦怠的实例。

第十一章　应激与挫折管理

从第十章中我们知道，情绪对个体的身心健康、工作质量有很大影响。然而，情绪并非无源之水、无本之木，它常常来源于工作和生活中的压力和挫折。人生在世，不如意事十之八九。面对林林总总的坎坷磨难，很多人会产生怨天尤人的哀叹，悲天悯人的愁思。这种负面情绪如果不能及时得到有效缓解，就会转化为一种无形的压力，弥散于心灵之上，让人无法呼吸。当这种状况扩散开来，波及工作、学习、生活等诸多方面时，挫折感就产生了。面对压力，应该怎么办？遭受挫折，应该怎样应对？本章拟对应激和挫折的本质、来源、影响因素进行分析，并对应激管理和挫折应对提出建议。

第一节　应激管理

一、什么是应激

应激（stress）也称为压力。这个概念与爱的概念有几分相似之处，每个人都知道这个概念所包含的意思，但是没有两个人会用同一种方式来定义它。当我们大多数人谈到压力时，通常指的是我们所感受到的来自周围的压力。学生谈到压力可能是因为考试成绩太差，或者是一篇重要论文的最后期限到了；父母谈到的压力可能是因为养家糊口而带来到经济负担；教师谈到的压力可能是因为一方面他们要保持自己在专业领域内的成就，另一方面还要把课堂教学做好；医生、护士与律师谈到的压力可能是因为他们要应付病人或当事人提出的无休止的要求。在当代的科学文献中，应激这个概念至少有三种不同的含义。

第一种，应激指那些使人感到紧张的事件或环境刺激。从这个意义上来讲，它对人是外部的。在本书中，我们倾向于将这种意义上的应激称为应激源（stressor）。

第二种，压力指的是一种主观反映，是紧张或唤醒的一种内部心理状态，它是人体内部出现的解释性的、情感性的、防御性的应对过程。

第三种，压力是人体对内部需要或伤害性侵入的一种生理反应。坎农（Cannon）和塞耶（Selye）都是在此意义上使用应激这一术语的。

以上三种含义各有侧重之处，但是心理状态和生理反应常常相伴而生，不可能单独出现，所以应激是一种伴随着外部刺激而产生的身心状态，既包括生理反应，又包括心理反应，既有遗传的成分，也有习得的成分。在应激状态下，人们往往会六神无主、寝食不安，身心发生剧烈变化。不管遭遇到何种应激事件，身体都会在紧急情况下自动产生一系列的反应，如血压升高、手心出汗、心跳加速等。应激引起的心理反应范围很广，既有认知上

的反应，如知觉范围变小、记忆功能失调、思维紊乱等，又有情绪上的反应如愤怒、害怕、沮丧、焦虑、悲伤等。

加拿大医生汉斯·塞耶（Hans Selye）首先使用“应激”一词来描述身体的生理反应机制，认为应激是身体对强加于自身的外部要求的非具体反应。他提出了“一般适应综合症”（general adaptation syndrome）的概念，用来指应激情境对人所造成的一般症状。图 11.1 给出了这些症状产生的一般模式。

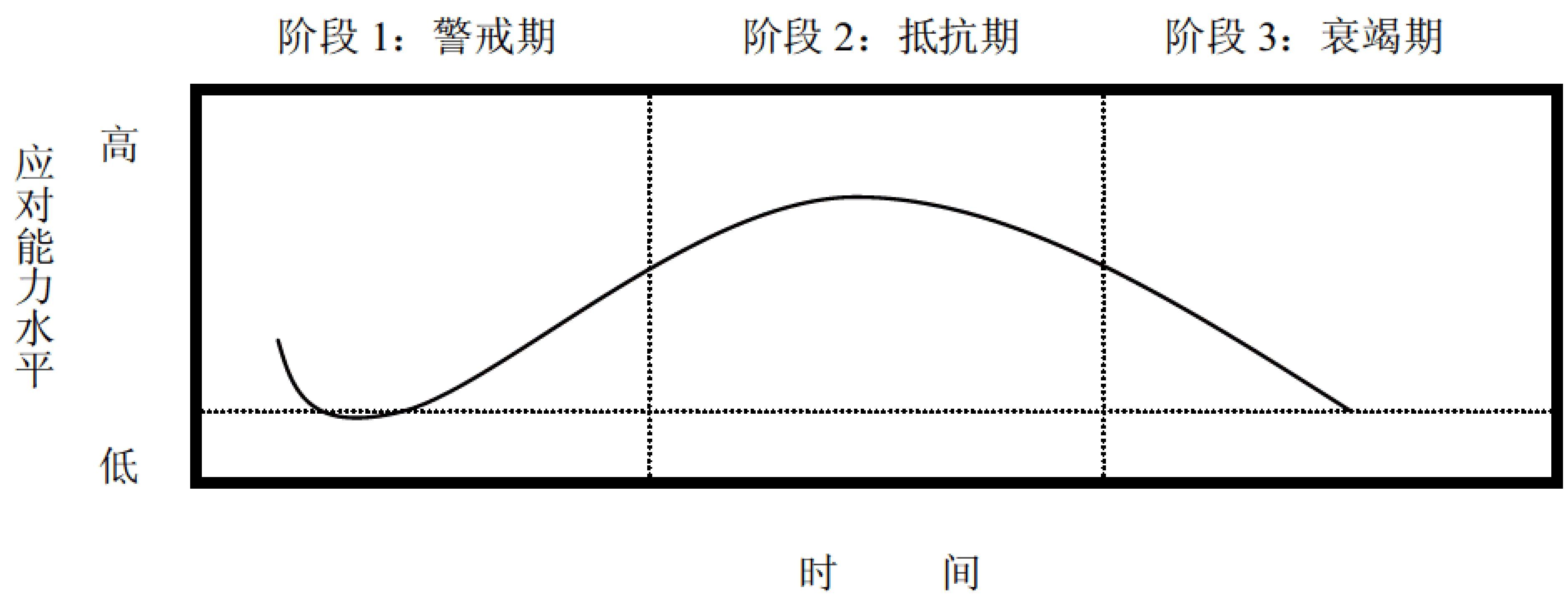


图 11.1　一般适应综合症模型

第一阶段为警戒期。刚出现应激事件时，警戒反应就会发生。个体处理该应激事件的水平会很低（低于平时的水平），感到手足无措，但是随后人体会迅速动员自身的心理资源，做出自我保护性的调节。如果调节有效，警戒就会解除，个体恢复到正常状态。

第二阶段为抵抗期。如果个体未能在第一阶段解除危机，致使应激源继续存在，那么个体将会试图调动更多的、更加重要的身心资源去排解压力。如果能够奏效，危机就可以在不伤害身体或生理健康的情况下得到解决，否则就会进入第三个阶段。

第三阶段为衰竭期。机体的身心能量是有限的，如果应激情境一直持续下去，个体的适应能力就会下降，身心能量就会耗尽，直至精疲力竭，陷入崩溃状态，身体上出现各种症状。

上述过程会因为个体的应激感受性和适应能力的差异而表现出不同的模式。一般说来，那些应激感受性较低且适应能力较高的人，其第一和第二阶段持续的时间比较长；而那些应激感受性偏高且适应能力偏低的人则很快就会到达最后阶段。也就是说，应激过程还会受个体因素的影响，存在个体差异。

二、应激源和应激感受性

引起应激感受的客观因素有很多，主要分为两类：一类是自然或物理事件应激源，如突如其来的恶劣天气、意外交通事故等。这类应激源持续的时间较短，且不会引起人们社会情绪方面的反应，相对比较容易应对。另一类是社会事件应激源。这类应激源又可大致分为两种：一种是与组织或工作相关的应激源，称为工作应激源；另一种是与组织或工作

无关的应激源，称为生活应激源。

社会事件应激源通常持续的时间较长，引起的社会情绪反应较为强烈而持久，应对起来比较困难，所以下面着重分析这类应激源及其产生作用的过程。

（一）工作应激源及应激感受性

工作场所中所发生的几乎任何事情都可能造成应激，所以工作应激源有多种形式。研究者们已经对与工作有关的应激源进行了总结，并探讨了它们经由个体应激感受性的调节产生作用的机制，形成了如图 11.2 所示的模型。

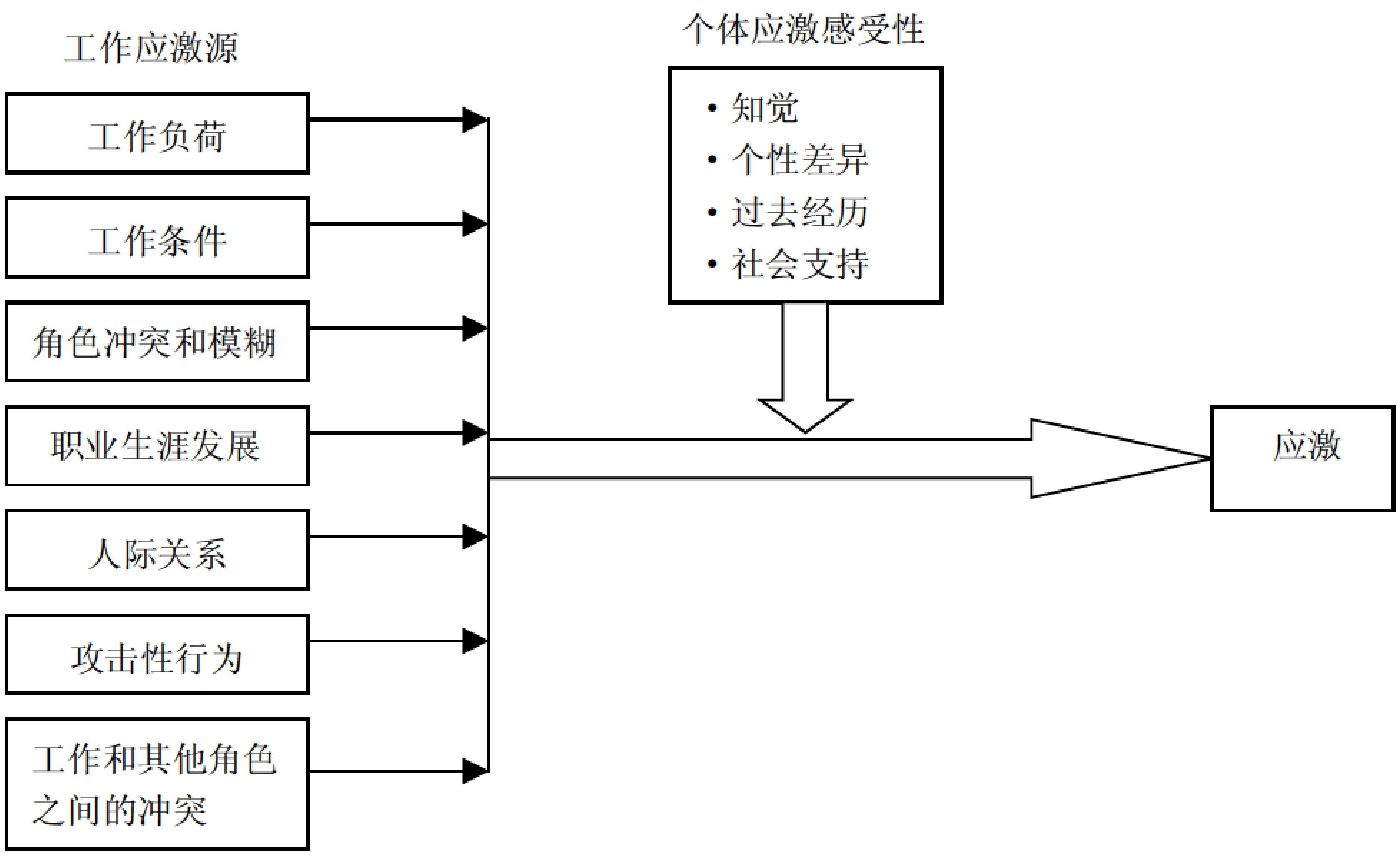


图 11.2　工作应激源及其作用机制

1．工作应激源

（1）工作负荷

对许多人来说，有太多的工作要做。如果没有足够的时间、资源或能力来完成他的工作，就会产生应激。这会让员工超负荷工作，加班加点，导致身心疲惫，降低工作效率。那么，工作太少、过于安闲又会怎样呢？研究发现，从忙碌中突然闲下来的人也会产生应激问题，例如，刚刚退休的人或从领导岗位上退下来的人，会感到很不适应。而且，长期处于闲置状态也常常构成应激事件，许多人都有过工作内容太少、过于清闲而感觉度日如年的体验。

（2）工作条件

恶劣的工作环境和条件是另一种重要的应激源。工作场所空间狭窄拥挤、温度过高或过低、噪音过大、光线太暗或太亮、辐射和空气污染等都会引起员工的应激。

此外，旨在改善工作条件的新技术的引进和使用也会带来应激。例如，越来越多的人患上了“电脑焦虑症”或曰“技术压力”。技术压力主要与几种忧虑有关：首先一些员工由

于不熟悉新技术或者掌握新技术有困难而感到忧心忡忡；其次，有些员工担心新技术的使用让他们丧失已建立起来的技术优势；第三，有些员工担心过于依赖电脑等新技术而荒废自己的技能，工作变得机械而无聊；第四，有些员工担心电脑等新技术会取代人，从而导致公司裁员；第五，有些员工甚至担心电脑等新技术会使管理者们对自己的控制更加严密，失去工作的自主性或自由空间。

（3）角色冲突和角色模糊

工作中，当不同的上司对同一个员工提出了不同的期望和要求，或者有多种相互冲突的岗位目标时，该员工就会产生角色冲突；当岗位职责不明确，员工不清楚自己应该做什么和怎么做时，就会产生角色模糊。绝大多数人不喜欢这样的不确定性，并发现这会给自己造成巨大的压力。事实上，在组织情景中，角色模糊现象非常普遍。一项有关角色模糊的调查发现，35%～60%的员工经历过某种程度的角色模糊。员工产生角色模糊的程度存在跨文化差异。一项由 21 个国家参与的调查研究发现，角色模糊在亚洲和非洲国家中出现的频率相对较低，这些国家中的权力距离较高，个人主义倾向较低，领导喜欢发号施令，提出明确的要求，员工较少思考和怀疑领导的意见，愿意或乐于按照领导的意志行事，不容易产生角色模糊。相反，在低权力距离、高个人主义的西方国家中，领导希望员工确定目标，明确责任，决定工作的方式，所以角色模糊现象相对严重。

（4）职业生涯发展

组织情景中与职业生涯规划和发展相关的应激源包括工作安全度、提升机会、培训和发展机会等。工作不稳定，随时担心被炒鱿鱼，或没有培训机会，缺少发展空间，都会给员工带来压力和应激。员工提升过慢（没按照期望的进度提升）或过快（提升到超过个人能力的工作岗位）都会感到压力和产生应激。

（5）人际关系

与上司、同事、下属愉快相处，保持良好的工作关系，是组织生活极其重要的方面，也是员工在组织中要实现的主要目标之一。复杂而紧张的人际关系、防不胜防的组织政治行为、“办公室政治”等是工作应激的主要来源，其对员工造成的压力常常超过工作本身带来的压力。

（6）攻击性行为

组织中的攻击性行为常以暴力侵害或性骚扰的形式出现。对员工造成实际的生理或心理伤害的攻击性行为被称为工作场所暴力。在工作场所受到暴力侵害或暴力威胁的个体更有可能体验到负面的情绪，并产生消极的行为反应，包括较低的生产率和较高的缺勤率。性骚扰是工作场所中第二种主要攻击性行为，通常是指不受欢迎的、有着性意味的接触和交流。有相当一部分女性员工在工作场所中受到过性骚扰。

（7）工作与其他角色的冲突

除了职业角色之外，人在生活中还扮演着许多其他角色。这些角色可能与工作发生冲突，成为应激源。例如，员工与家庭成员呆在一起的愿望可能与他必须工作以促进职业生涯发展的愿望相冲突。目前，越来越多的双职工夫妇，已将工作和家庭角色冲突引向尖

锐化。

2．应激感受性

工作应激源只是造成工作应激的客观条件。它们是否带来应激或造成的应激水平有多高，还与遭遇它们的个体的应激感受性有关。应激感受性是指个体对应激源或应激事件的敏感程度。有些人应激感受性高，周围环境有一点风吹草动，就会引发他的应激状态或反应，这类人易受应激事件的影响，产生较高的应激水平，而另一些人则相反。应激源就是经由个体应激感受性的调节而最终产生应激状态的。

一个人应激感受性的高低既与他的个性和心理特点有关，也与他的生活经历和所处的环境有关。具体地说，个体对特定事件的应激感受性主要受以下四个因素的影响：

（1）人对情景的知觉。员工对应激事件的知觉可能影响到他如何体验应激事件。例如，两个员工同样面对工作职责的巨大变化。一个员工可能会觉得新职责是学习新技能的机会，并把这种变化知觉为上司对自己工作能力的肯定和信任。相反，另一个员工可能把这种变化知觉为一种威胁和上司对自己的不信任。

（2）个性差异。个体在动机、态度、个性和能力方面的差异也会影响员工是否体验到工作应激和对应激事件的反应强度和方式。

（3）过去经历。一个人在特定环境中感受到的应激程度依赖于他对该环境的熟悉程度或该环境与他所经历过的环境的相似程度，同样地，当一个人遇到一件突如其来的事情时，他所感受到的应激水平取决于他以前是否经历过类似的事情。例如，刚刚走上工作岗位的新员工或刚来到一个新单位工作的人，常常会体验到更多的应激。经验或阅历可以降低一个人的应激感受性，培训或训练也可以使新员工尽快地适应新环境，降低应激感受性。

（4）社会支持。一个人遇到应激事件时，实际得到的社会支持或得到社会支持的可能性大小，将会影响他的应激感受。面对同样的应激源，孤立无援的个体显然要比受到多方援助的个体承受的压力更大。

（二）生活应激源

值得注意的是，工作应激源并非影响工作应激的唯一来源。发生在工作场所以外的社会生活事件产生的持久应激也会被带到工作中来，从而影响工作应激。因此，有必要对工作应激源之外的其他社会生活应激源加以介绍和分析。

“天有不测风云，人有旦夕祸福。”生活应激源形式多样，内容各异，但大体上可以区分为以下两类：

1．重大生活事件

人们在生活中可能会遇到一些突发性的或重大的生活事件，如离婚、结婚、亲人亡故等。这样的事件会对人们的身心健康产生什么影响呢？这个问题引起了健康心理学家们的广泛注意，开展了大量的生活事件研究，试图弄清楚生活事件、应对方式和身心健康之间的关系。表 11.1 列出了有关研究所确认的重要生活事件及其应激指数。应激指数从 1 到 100，分数越高表明该事件引起的应激水平越高。

表 11.1　各种生活事件及其应激指数

事　　件	应 激 指 数	事　　件	应 激 指 数
配偶死亡	100	怀孕	40
离婚	73	密友死亡	37
夫妻分居	65	孩子离家出走	29
入狱	63	与配偶、家属有麻烦	28
家庭成员死亡	63	与老板发生矛盾	23
受伤或生病	53	搬迁	20
结婚	50	放假	13
被解雇	47	圣诞节	12
退休	45	轻度违法	11

资料来源：Holmes,T.H., & Rach,R.H.（1967）

2．日常生活事件

日常生活中充满了无数令人烦恼的小事。虽然这些事情的应激指数不高，但发生的频率很高。这些事件称为日常生活事件。它们可能发生在生活的各个方面，包括家务的烦恼、时间压力烦恼、经济烦恼等。人们经常遇到的应激事件都属于此类应激事件。许多研究表明，人们经历的日常应激事件的频率与自我报告的应激水平之间存在相关。

三、应激的作用与后果

应激是工作生活不可分割的一部分，而且对人的工作和生活造成的影响并非全是负面的和消极的。适度的应激可以让人处于正常的唤醒、紧张状态，保持一定的张力，这对人们的身心健康及正常工作和生活不仅是有益而无害的，而且是必要的。在一定意义上，可以说生活就是应激。

然而，过度的或不适当的应激却会带来严重的身心后果，进而影响工作效率和生活质量。下面从三个方面着重阐述应激的负面影响。

（一）应激与身体健康

过度应激的直接生理后果是血压增高、心率加快、出汗、呼吸急促和肠胃活动紊乱等。有证据表明，个体的身体健康和其承受的应激之间有着非常紧密的联系。一些专家估计，应激与 50%～70%的各种各样的生理疾病有关，其中有些是严重的、威胁个体生命的疾病，如心脏病、中风、溃疡、头痛、糖尿病以及癌症等。许多研究表明，高水平应激增加了患某些传染病的概率，如上呼吸道感染以及各种细菌性传染病。总之，应激常常对个体健康带来消极影响。与应激有关的疾病给个人和组织都带来了巨大的精神压力和经济负担。

（二）应激与心理健康

过度应激的心理后果包括愤怒、焦虑、抑郁、较低的自尊、较差的智力功能等。当人们长期处于应激状态而无力自拔时，他们的精神就会被拖垮，出现各种各样的心理症状。上一章所介绍的工作倦怠就是组织情景中个体长期处于应激状态导致的一种综合性心理症状。

一般而言，需要经常与人打交道的职业产生的应激频率和强度较高，最容易产生工作倦怠。图 11.3 按照与人接触的频率和强度两个维度，对不同职业进行了归类，并指出了这些职业可能诱发倦怠的程度。

人际接触频率 \ 人际接触强度	低	高
高	招待员 销售代表 图书管理员 救济代表 （中度倦怠）	社会工作者 客户服务代表 学校教师 护士 （高度倦怠）
低	研究物理学家 森林巡逻员 炼油厂操作员 实验技工 （低度倦怠）	护理人员 公共辩护人 消防队员 警探 （中度倦怠）

图 11.3　根据人际接触的频数和强度来预测工作倦怠水平

（三）应激与工作绩效

过度应激还有很多行为后果，如退缩、逃避、撒谎、攻击等。在组织情境中，过度应激常常导致不按时上班、缺勤、消极怠工、离职等行为，严重的还可能表现为破坏或盗窃公司物品等，从而降低个人的工作绩效，阻碍组织目标的实现。

那么，组织情景中员工应激水平过低又会怎样呢？图 11.4 详细地描述了应激水平与工作绩效的关系。纵坐标表示从低到高变化的工作绩效水平，横坐标表示从低到高变化的应激水平。它表明：过低和过高的应激都未达到最佳的绩效水平，只有在适度的应激水平上，员工的绩效才达到最高。在低应激条件下，员工的警觉性低，缺乏挑战性刺激，处于麻木不仁的状态，不能发挥他们最好的水平，而在高应激条件下，员工唤醒水平过高，环境太具有威胁性，处于疲于应付状态，也不能尽力于工作。

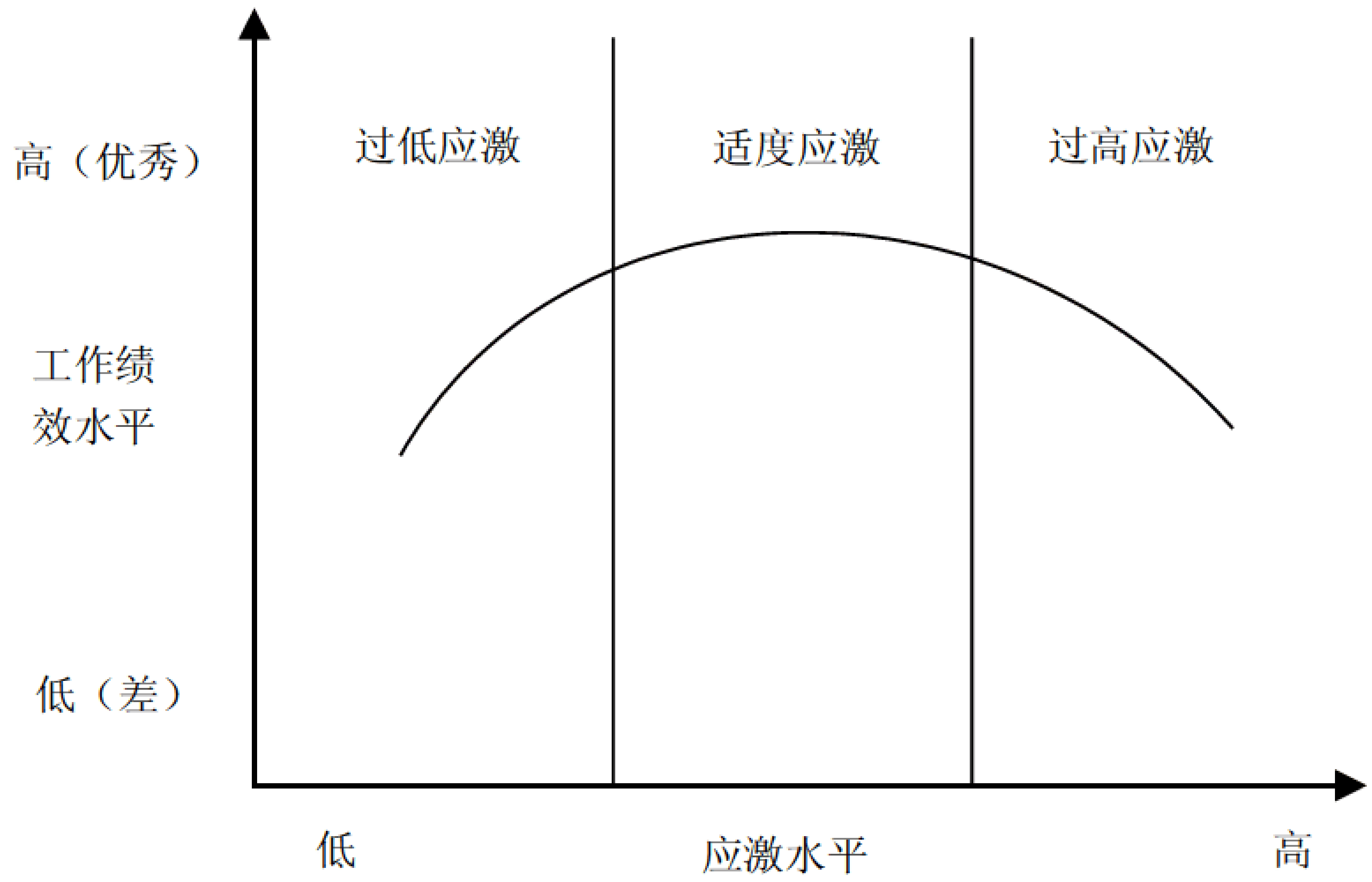


图 11.4　应激与工作绩效的关系

“刚好够就可以了，不要太多”

肖琳是一家公司的员工。在过去的几个月里，她向公司董事会递交了三份重要报告。第一份报告彻底失败。她是如此紧张，以至于实际上对发生的事情记得很少。报告返回来了，有各种各样的问题，一个比一个难以解答。董事会成员最终同情她，停止了质询。后来，马丽——她的老板——帮助肖琳分析她的作业水平。她们确信，除了两个问题之外，肖琳实际上知道问题的答案。之所以发生这些事情是因为她太紧张了，不能清楚地思考。马丽对她的建议是：“你不得不为自己制造工作压力，但你不能有太多的压力，以至于不能正确思考。”

肖琳在马丽的建议之下，仔细准备了下一份报告。她做了一份高质量的报告，并自信地回答了董事会的问题。董事会热情洋溢地表扬了她。

现在，肖琳带着她的第三份报告来到董事会。她显得很“平静”，她一点也不紧张。事实上，就在她来董事会办公室前，她还在为另一项工程工作。虽然她的报告比第一份要好，却远远没有达到第二份的水平。“怎么啦？”她想知道。在马丽的帮助下，她再次试图弄清自己的问题。经过讨论，她们知道第一份报告失败是由于应激水平太高，具有讽刺意味的是，这一份报告的失败却是因为应激水平太低。

管理者常想知道他们自己和下属最恰当的应激点在哪里，但是很难准确回答这个问题。例如，员工经常缺勤的原因可能是工作过重（应激水平太高），也可能是厌倦（应激太少）。因此，图 11.4 中的曲线会因不同的人或任务而呈现不同的形状。也就是说，不同的人、在不同的任务上，所需要的最佳的应激水平是不同的。

四、应激管理

当个人或组织感到应激可能或已经威胁自己的身心健康和正常工作效率时，应激就成

了需要管理的问题。应激管理就是个人或组织采取策略和方法来应付和处理应激问题的过程。

下面从常用的应激管理策略和当今企业中比较流行的EAP的理论和实践两个方面阐述应激管理问题。

（一）应激管理策略

1．消除应激源

解决应激问题最有效的途径就是消除应激源。有好几种方法有助于消除工作场所中的应激源：（1）授权给员工，这样员工就会对他们的工作和环境有更多的控制，减少来自上司的压力造成的应激。（2）通过更加有效地选拔和安置员工，使他们的能力与工作要求相匹配，就可以降低与工作相关的应激。（3）噪音和安全隐患也是一种应激源，改善这些条件也可以降低工作中的应激。

员工在消除应激源的过程中也起着重要作用。例如，如果员工的应激体验是由于模糊的角色预期，他就要努力从他人那里获得更多的信息来区分这些预期。如果一项工作太具有挑战性，员工可以把它分成一系列更小的任务，降低它的挑战性。员工还可以通过提高人际技巧来降低工作场所中的暴力伤害。

2．远离应激源

消除应激源可能是一种理想化的策略，很多情况下并不可行，因为你根本无法消除应激源。远离应激源是另一种应激管理的策略。它有两种：一种是永久地远离应激源，例如把员工调换到更适合他的工作岗位上去。如果没有合适的岗位，就让员工离开组织。另一种是暂时远离应激源。例如在连续工作一段时间后实行集体放松，要求员工集体做深呼吸，也可以做一些肢体伸展运动，缓解肌肉张力，加速血液在体内的循环，帮助把氧气输送到大脑，安排下午茶、水果等，让员工暂时远离紧张的工作环境。休假是另一种暂时远离应激源的方法。许多公司都采用带薪休假的办法让员工储存新的能量以更好地应对工作中的压力。

有一个压力管理的小故事很能说明暂时远离应激源的策略：

培训师在课堂上拿起一杯水，然后问台下的听众：“各位认为这杯水有多重？”有人说是半斤，有人说是一斤，培训师则说：“这杯水的重量并不重要，重要的是你能拿多久？拿一分钟，谁都能够；拿一个小时，可能觉得手酸；拿一天，可能就得进医院了。其实这杯水的重量是一样的，但是你拿得越久，就越觉得沉重。这就像我们承担着压力一样，如果我们一直把压力放在身上，到最后就觉得压力越来越沉重以至于无法承受。我们必须做的是放下这杯水，休息一下后再拿起这杯水，如此我们才能拿得更久。所以，各位应该将承担的压力在一段时间后适时地放下，好好休息一下，然后再重新拿起来，如此才可承担更久。”

3．改变对应激的知觉

面对同样的情景，员工经常体验到不同的应激水平，因为员工对情境的知觉是不同的。因此，可以通过改变对情境的知觉降低应激水平。这并不是说应该忽视风险或其他应激源，而是说要增强个体的自我效能感和自尊，强化自我，以至于不再把工作的挑战看成是一种

威胁。通过积极的自我暗示，可以改变对应激的知觉，把坏事变成好事。乐观的思维习惯一旦形成，就会使人保持愉快的心境，即使工作很忙，也不会感到紧张，而是感到活的很充实、有自信。

4．控制应激后果

另一种应对工作应激的方法就是控制应激的后果。一种最常见的办法就是健身运动。大多数的美国公司都鼓励员工使用这种应激管理策略。他们要么在公司内部建立健身中心，要么为在公司外健身的员工提供经济支持。体育锻炼有助于员工降低他们的肌肉紧张、生理节律等，因此可以减少应激造成的不良生理后果。

另一种控制应激生理后果的办法是放松和冥想。放松可以化解脑力和体力的紧张，让人摆脱烦恼和焦虑，肌肉松弛，内心宁静。一般的放松程序是：（1）选择舒适的位置；（2）闭上眼睛；（3）放松肌肉；（4）监视你的呼吸；（5）当思维浮现时，保持积极态度；（6）持续一段时间（如 20 分钟）。这些方法能有效降低人们的心跳频率、血压、肌肉紧张和呼吸频率。

5．接受社会支持

接受来自上级、同事、家庭、朋友和其他人的支持是一种更加有效的应激管理策略。社会支持可以提供给员工情感上或信息上的帮助，缓解应激感受。

社会支持至少在三个方面可以降低应激感受：首先，社会支持可以让员工觉得他们自己是有价值的。这就可以提高员工的自尊，增强处理应激源的信心。第二，社会支持可以提供信息帮助员工去解释、理解甚至消除应激源。第三，来自他人的情感支持可以直接缓解员工的应激体验。

（二）EAP 的理论与实践

上面介绍的五种应激管理策略大多是员工个人可以采用的策略。有时候员工个人不能有效地处理应激事件，需要来自组织的帮助。一项叫做员工援助计划（Employee Assistant Program，简称 EAP）的服务正在国外许多企业中盛行，并且正在渐渐为我国企业所接受。

1．EAP 的概念和分类

EAP 是企业为员工设置的一套系统的、长期的福利与支持项目。它通过专业人员对组织的诊断、建议和对员工及其家属的指导、培训和咨询，帮助员工解决自我及其家庭成员的各种心理和行为问题，从而提高员工在企业中的工作绩效。完整的 EAP 可以分成三个部分：第一部分针对造成问题的外部压力源，即减少或消除不适当的管理和环境因素；第二部分是处理压力所造成的反应，即缓解情绪、行为及生理等方面的症状；第三部分是改变个体自身的弱点，即改变不合理的信念、行为模式和生活方式等。如今，EAP 已经发展成一种综合性的服务，其内容包括压力管理、职业心理健康、裁员心理危机、灾难性事件、职业生涯发展、健康生活方式、法律纠纷、理财问题、饮食习惯、减肥等各个方面。解决这些问题的核心目的在于使员工在纷繁复杂的个人问题中解脱出来，减轻员工压力，增进其心理健康。

根据不同标准，对 EAP 可以进行如下分类：（1）根据实施时间长短，分为长期 EAP 和短期 EAP。EAP 作为一个系统项目，应该是长期实施的，但有时企业只在某种特定状况

下才实施员工帮助，例如并购过程中由于业务再造、角色变换、企业文化冲突等导致压力和情绪问题；裁员期间的沟通压力、心理恐慌和被裁员工的应激状态。又如，在空难等灾难性事件中，部分员工的不幸会导致企业内悲伤和恐惧情绪的蔓延。这种时间相对较短的员工帮助措施能帮助企业顺利渡过一些特殊阶段。（2）根据服务提供者的不同，分为内部EAP和外部EAP。内部EAP是在企业内部，配置专门机构或人员，为员工提供服务。比较大型和成熟的企业会建立内部EAP，其优点是能更贴近和了解企业及员工的情况，更及时有效地发现和解决问题。外部EAP由外部专业EAP服务机构操作。企业需要与服务机构签订合同。一般而言，内部EAP比外部EAP更节省成本，但员工由于心理敏感和保密需求，对工作人员的信任程度可能不如外部EAP。专业EAP服务机构往往有广泛的服务网络，能够在全国甚至全世界提供服务，这是内部EAP所难以企及的。在实践中，内部和外部EAP往往结合使用。

2．EAP的实施程序

EAP实施程序通常包括以下几个步骤：

第一，把脉与诊治。由专业人员采用专业的心理健康评估方法评估员工心理生活质量现状及其问题产生的原因。针对造成问题的外部压力源本身去处理，即减少或消除不适当的管理和环境因素。

第二，宣传与推广。搞好职业心理健康宣传，利用海报、自助卡、健康知识讲座等多种形式，引导员工对心理健康形成正确认识，鼓励其遇到心理困扰问题时积极寻求帮助。

第三，改善环境。一方面，改善工作硬环境——物理环境。另一方面，通过组织结构变革、领导力培训、团队建设、工作轮换、员工生涯规划等手段改善工作的软环境，在企业内部建立起支持性的工作环境，丰富员工的工作内容，指明员工的发展方向，消除问题的诱因。

第四，全员培训。开展员工和管理者培训，通过压力管理、挫折应对、保持积极情绪等一系列培训，帮助员工掌握提高心理素质的基本方法，增强对心理问题的抵抗力。

第五，心理咨询。组织多种形式的员工心理咨询，对于受心理问题困扰的员工，提供咨询热线、网上咨询、团体辅导、个人面询等多种形式的服务，改变个体自身的弱点，即改变不合理的信念、行为模式和生活方式等。

3．实施EAP的意义

第一，企业通过EAP，可以维持员工的身心健康，减少企业医疗保健费用的支出，降低离职缺勤率，节省招聘费用，从而降低成本。据英国专家研究显示，每年由于压力造成的健康问题通过直接的医疗费用和间接的工作缺勤等形式造成的损失达整个GDP的10%！正是在这种情况下，企业纷纷寻找出路，通过EAP维持员工的身心健康来抑制医疗费的增加。事实证明，EAP为企业创造了可观的效益。据美国健康和人文服务部1995年的资料，在美国，对EAP每投资1美元，将有5～7美元的回报。Marsh & Mclennon公司曾对50家企业做过调查，在引进EAP之后，员工的缺勤率降低了21%，工作中的事故率降低了17%，而生产率提高了14%；1990年McDonnell Douglas对经济增长的研究报告显示，实施EAP项目4年来，共节约成本510万美元；美国一家拥有7万员工的信托银行引进EAP之后，仅仅一年在病假的花费上就节约了739870美元的成本；摩托罗拉日本公司在引进

EAP 之后，平均降低了 40%的病假率。

第二，企业通过 EAP 提供健康咨询活动及各种健康服务，可以培养员工积极健康的生活方式，提高员工满意度，提升劳动生产率。人是竞争的第一要素，员工的身心健康是企业的最大财富，并且企业发展对员工在体能、智能和心理素质等方面也提出了更高的要求。为此，要求企业提供健身设施，鼓励员工参加体能训练，给员工提供营养配餐及帮助员工克服生活上的一些不良习惯。由于员工的工作业绩与身心状态有密切的关系，并且组织环境、安全保障程度、人际关系、个人前途等，都会对员工工作绩效产生影响。因此，EAP 服务提倡关心人、爱护人、激励人、安慰人，为员工创造一个和谐的工作环境，以此提高员工满意度及增进组织的工作绩效。

第三，EAP 服务有助于建立“以人为本”的积极健康的企业文化。企业文化是企业发展的精神动力，它的核心是对人的重视和尊重。EAP 不仅强调员工的心理因素，而且把全面关心员工的身心健康和机能的正常发展作为目标，EAP 不只是把员工看成是管理的对象，更重要的是作为伙伴和朋友，强调采用体贴、关怀的方式构筑企业和谐的气氛。EAP 不仅注意减轻员工在企业的压力（如提供升职机会、安全保障等），而且努力帮助员工解决企业之外的压力（如协助员工购房、教育子女等）。EAP 不仅重视人力资源管理方面的软件开发（如激励、诱导等），而且重视与人力资源管理有关的硬件开发（如兴建健身设施、提供服务项目等）。EAP 服务可以使员工产生对企业的认同感，增强员工对企业的忠诚度，从而使企业员工在心理上产生一种凝聚力。

第二节　挫折管理

小王是一家化妆品公司的财务副主管，性格开朗，对工作充满热情，富有上进心，与同事相处也十分融洽，被认为是现任财务主管的接班人，他对此也充满信心。但事与愿违，上级任命了另一位高学历的年轻人为财务主管，从那以后，小王整个人似乎发生了转变，开始嗜睡和酗酒，有时还会无缘无故冲同事发火。

当小王升迁的梦想破灭时，他产生了强烈的挫折感。在工作场所中，除了前面提到的应激和压力之外，挫折也会对员工的健康和工作绩效产生严重的负面影响。在本节中，我们将考察挫折的本质和影响，以及如何进行挫折管理。

一、挫折的本质及表现形式

（一）挫折的本质

在实际的工作中，员工甚至组织不总是一帆风顺的，经常会遇到挫折（frustration）。那么什么是挫折？挫折通常可以理解为失败、失意、遭遇困难等意思。从心理学的角度看，就是员工在实现其目标的过程中，遭遇到种种的干扰和障碍，导致他们无法实现目标而出

现的消极情绪状态。简单地说，挫折就是有目的的行为受到了限制。 仔细分析这一定义，不难看出，它有两方面的含义：首先，挫折是一种客观的条件或状态，这种条件或状态的存在阻挠了个体预设目标的实现或干扰了正在进行的工作；其次，它是指由于目标受阻而引起的内在消极情绪状态，即我们通常所说的“挫折感”。

（二）挫折的表现形式

挫折有两种表现形式：外部性挫折和内部性挫折。外部性挫折主要是指由于外部因素的障碍和干扰而形成的挫折；内部性挫折则是员工的主观原因引起的挫折。下面分述两种挫折的具体表现形式。

1．外部性挫折

（1）条件性挫折：由于缺少外部条件的支持或强化致使目标无法实现而产生的心理挫折。有人也将其称之为强化延迟。例如，工作中有上佳表现的员工迟迟得不到上级领导或主管部门的肯定；由于家庭或其他方面原因造成员工绩效不佳而长期得不到领导或上级的关心和信任。

（2）丧失性挫折：原来一直得到满足的需要突然丧失或者部分丧失而产生的心理挫折。有人把它称为持续强化的中断。例如，一直受到表扬的员工突然不再得到表扬；一直被委以重任的员工不再被重用。

（3）干扰性挫折：由于外力干扰或阻碍致使无法实现预期目标而产生的挫折。由于父母、小孩生病而造成员工工作绩效的下降或者由于交通问题导致员工总是迟到等都会使员工产生干扰性挫折。

2．内部性挫折

（1）生理性挫折：个体生理上的某些不足或缺陷而引起的心理挫折，其中包括一些个体自身想象的缺陷。例如，有的员工认为自己的五音不全，致使在公司演讲或举办晚会时不能表现自己而产生心理挫折。特别是 2003 年出现的张静整容事件，更使有的员工注意自己的容貌，甚至出现埋怨自己太矮、太胖等现象。

（2）强烈的反应障碍：突发的个体需要与当前个体所处的状态产生矛盾而形成的挫折。例如，正在忙于工作的员工，突然产生了某种强烈的个人需要，但他又无法撇开工作去满足自己的需要。

（3）冲突性挫折：主要是指个人动机之间冲突而产生的挫折。例如，双趋冲突，既想在人力资源管理部工作，又想去到财务部工作；趋避冲突，既想做车间主任，又不想努力工作；双避冲突，既不想受罚，又不愿意积极工作。

（4）怀疑性挫折：纯粹因为个人的主观怀疑而设置的心理障碍形成的挫折。例如，总是怀疑领导不信任自己，怀疑同事在背后议论自己，怀疑自己做的工作是否有价值等。

二、挫折的来源

托尼早上赶去上班，可是发生交通堵塞，到单位时已经晚了半个小时，为此经理狠狠

地批评了他。怀着郁闷的心情，托尼开始了一天的工作。这样一系列的事件组成了对托尼的挫折。那么，挫折的来源究竟有哪些呢？从企业管理的角度来看，可以这样来划分挫折的来源：

（一）工作挫折

（1）角色模糊：当经理要求员工工作，但未能使其清楚地理解岗位职责和希望产生的绩效，从而导致任务未顺利完成时，员工往往产生强烈的挫折感，因为他们不知道要做什么、怎样做才能达到经理的要求。刚进入公司不久的员工，工作热情通常都很高涨，但经过一段时间就可能变得松懈，除了个性因素外，工作角色模糊是造成这种情况的主要原因。

（2）人际关系：与同事的工作关系不融洽同样会使员工产生挫折感。麦克利兰（McCelland）认为：人们都有交往的动机，即希望与他人建立并维持良好的关系，当这种动机遭遇障碍时，人们往往会体验到强烈的挫折感并对随后的行为产生不良的影响。很难想象，在企业中，特别是在一个团队中，常常与其他成员发生矛盾的员工能很顺利地完成任务并保持长久的良好工作态度。

（二）非工作挫折

每周 168 个小时，而多数员工只工作 40～50 个小时。他们在非工作时间内遭遇的挫折同样可以带进工作场所。非工作挫折的主要来源是家庭、健康和人格因素。

（1）家庭：和睦的家庭是员工安心工作的前提和保障，家庭发生问题可以成为员工主要的挫折来源。越来越多的研究关注“工作—家庭冲突”以及这种挫折给员工带来的负面影响及解决方法。试想，在清晨出门前与妻子发生争吵后，还能保持良好的心情去公司上班，并顺利地开展一天的工作？

（2）健康：无论是员工自己，还是家人朋友患病或受伤，都会使员工产生挫折感和焦虑。一个告知其家人生病住院的电话会使原本聚精会神工作的员工产生强烈的挫折感，难以安心地继续工作下去。另外，如果在太短时间里经历过多的挫折，就会发生与挫折有关的健康问题。而健康问题本身就是挫折的来源。

（3）人格：弗里德曼（Friedman）将人格分为 A 型和 B 型。因其个性特征和工作理念不同，A 型人格的员工对待挫折事件的方式与 B 型人格的员工也会有所不同。斯塔波（Strube）和伯兰德（Boland）的研究表明：因为他们对自己的能力很自信，也更加努力，所以当目标未实现时，A 型员工比 B 型员工更容易体验到强烈的挫折感。

三、员工受挫后的行为表现

被经理批评后，托尼的心情很不爽，满腹牢骚地开始工作。午餐的时候，有个同事给他开了个不大不小的玩笑，平时托尼会一笑了之，但今天他无法控制自己的情绪，与那位同事发生了争执。事后托尼很后悔，但他感觉当时确实无法控制自己，这是为什么呢？

当员工遇到挫折后，必然会有一定的行为表现。由于个人的生活经历和个性不同，行

为表现也会各种各样。这些表现可分为理智性的和非理智性的。下面主要讨论一下非理智性的行为反应。

1．攻击行为

根据约翰·多兰德（John Dollard）的“挫折-攻击”理论，员工遇到挫折的时候，自然会产生不满的情绪。当这种情绪发展到愤怒的地步时，就可能对阻碍满足自己需要的障碍做出反抗，形成攻击行为。那么，什么时候员工最容易产生攻击行为呢？这取决于以下四种条件:（1）受挫折的强弱；（2）受挫折的范围；（3）受挫折的频率；（4）攻击反应可能受到惩罚的程度。此外，员工对挫折的归因也会直接影响其是否产生攻击行为。当员工认为是他人或组织造成挫折事件的发生，特别是故意造成的时候，愤怒的情绪就会产生，相应的攻击行为就会出现。员工的攻击行为可以分为直接攻击与转向攻击两种。直接攻击是把矛头指向造成其挫折的人或物；转向攻击是把攻击的矛头进行转移，指向与挫折原因无关的目标。例如，托尼无法向经理发火，于是将愤怒发泄到开玩笑的同事身上。

2．退化行为

根据阿德弗（Alderfer）的 ERG（生存-关系-成长）理论，退化反应是指人在遭受挫折后，做出与其年龄不相称的幼稚行为，似乎又回复到儿童时期的习惯与行为方式。有的人遭受挫折后大哭大闹、撒泼打滚，有的人受挫折后盲目地追随和相信别人。例如，一个平时有主见的员工在与上级管理者产生矛盾后，很可能听信那些关于自己上级的负面消息，矛盾越深，这种盲从的可能性就会越大。从职业生涯的角度看，一个人受到挫折，也可能从一定层次的职业阶梯位置下行，去从事那些简单、初级、低层次的工作，而不能做到职业维持和前进。

3．冷漠态度

冷漠是指个人受到挫折后不以愤怒和攻击的形式表现，而代之以一种貌似无动于衷的冷淡态度。当一个人在职业中受到挫折、又无法脱离这种工作时，往往会产生冷漠的反应，其结果是对工作丧失热情，以至消极怠工。根据弗洛伊德（Frued）的防御机制理论，受挫折者绝不是没有心理上的不满和愤怒情绪，只是将这种情绪反应暂时压抑下去，在外部行为上表现出对造成自己挫折的对象漠不关心、沉默冷淡的样子。冷漠反应与受挫程度、心理承受力和自信心强弱等因素有关。一般而言，性格内向，心理承受力较弱，自信心不强的员工更容易以冷漠行为应对挫折。

4．固执反应

固执是指人受到挫折以后执意地重复某些没有目的的活动。在大多数情况下，这些重复性活动是没有效果的，是在作无用功。具有固执反应倾向的员工往往缺乏机敏的品质与随机应变的能力，他们找不到合理的解决问题的方法，只能不断地重复过去的活动以图减轻心理上的焦虑。

5．妥协反应

员工受挫后，有时会采取妥协的方式来减轻心理的紧张状态。费斯廷格（Festinger）的“认知失调”理论认为，当人们遇到的情况与心里的想法不一致时就会产生认知冲突，最常见的解决办法就是被迫服从，之后再设计合理的理由去解释，以减轻失调的紧张状态。具体来说，妥协反映有以下几种形式：（1）文饰作用：文饰作用是自我安慰，是指人们受

到挫折后会想出各种理由原谅自己或者为自己的失败进行辩解。这非常类似于平常所说的“阿Q精神”。一个学生考试成绩很差，会用“我不作死啃书本的书呆子”为由安慰自己。（2）投射作用：一个人把自己身上存在的不良品质强加于别人身上。投射作用是一种无意识的反应，例如，一个对别人抱有成见的人会认为别人对自己也有成见。自己内心深处有贪污动机的人，却常常宣扬别人常常收红包。有的心理学家认为被害妄想是存在于自己内心的攻击行为的一种投射。（3）替代作用：当一个人确立的目标与社会的要求相矛盾，或者受到条件的限制而无法达到时，就会设置另一个目标取代原来的目标，这就是替代作用。升华是替代作用的一种主要表现形式。例如，人们将攻击等不为社会所接受的动机，转移到较高境界的艺术或运动上。替代作用又可称为“补偿作用”，即一个人从一种活动中遭受了挫折，就从另一种活动中谋求成功，以求得补偿。补偿是具有积极作用的，是个人扬长避短，通过正当努力来战胜挫折的过程。（4）反向作用：人们表现于外的行为或情感与他们内心的感受完全相反，称之为反向作用。反向作用往往是为了掩盖内心憎恨、敌视的情感。例如，过分的亲切与屈从，背后可能掩藏有憎恶与反抗的动机。（5）表同作用：表同作用是与投射作用完全相反的表现。投射作用是把自己不良品质强加到别人身上，而表同作用则是把别人具有的、让自己感到羡慕的品质加到自己身上。这往往表现为模仿别人的言行举止，以别人的姿态风度自居。例如，有人把自己和在某项事业上获得成功的人物或有名望的集体单位联系在一起，从而求得一些间接的荣耀，借此减少挫折的影响。

6．自我惩罚

有些员工遭受挫折后，得不到周围人的支持和帮助，可能会认为自己一无是处，产生万念俱灰的感觉。这种员工往往将自己作为发泄的对象，进行自我惩罚。这可以看作是一种变相的攻击行为，只不过对象不是他人而是自己。本节开始案例中的小王就是以酗酒作为应对挫折的方法。

四、挫折的积极和消极作用分析

（一）挫折的积极作用

一提到挫折，人们首先想到的就是怎样采取措施克服员工的挫折感。那么，是不是挫折一定会降低员工的工作积极性呢？让我们先来看一个现实的案例：约翰是一家医药公司的销售代表，凭着他出色的能力、经验以及游刃有余的社会关系，公司的销售额非常可观，领导以及同事给予他极大的信任和支持，认为只要有约翰在，我们的腰包就会越来越鼓。可是市场就是这样残酷，在激烈竞争条件下，原本乐观的他们一下子陷入了泥潭，公司工人的工资都成问题。这使约翰在公司的地位一落千丈，公司的同事以及领导对他产生了疑问：他真的能行吗？约翰自身也感觉到了巨大的压力，为什么出现问题？他进行了自我归因，是产品还是我的销售战略出了问题？他决定走出去，来了解当前大众的情况。他设计了几个问题让消费者回答：（1）以前你喜欢某某公司的产品？为什么？（2）现在喜欢哪一家公司的产品？为什么？通过这几个问题，他分析自己产品的劣势，吸取了其他公司营销的优势，改进了公司策略，最终带领公司走出困境。从这一案例可以看出，挫折并没有使

约翰气馁，而是更加促进他努力工作，取得更加优异的成绩。

可见，挫折是有积极意义的，它的存在并不一定会降低员工的工作动机，相反适当适时的挫折还会对员工产生积极的影响。这主要体现在以下几个方面：（1）挫折增进了员工的适应性。只有遇到挫折时，员工才会重新审视自己过去的经验和方法，从自身找出原因，改变对现实不恰当的态度，更好地适应环境；（2）挫折可以增加员工行为的色彩或复杂性。员工如果从不遇到挫折，他们的行为模式就永远不会改变，工作时就象一台机器而不是一个人，生活就会缺少激情。只有当老一套的工作程序或方法遭遇了挫折后，员工才会试验或探索新的方法，进行技术或理念的革新；（3）挫折常常可以转化为行为的动力。“有压力，才会有动力”，当员工遇到挫折时，才会产生心理压力，迫使他们采取有效的措施来减轻压力，化挫折为动力。这启示管理者，在实际的管理活动中，可以适当适时地安排一定的挫折情境，激发员工的工作热情。

（二）挫折的消极作用

挫折的积极作用是有条件的。一般地说，只有当挫折的持续时间较短，个体有足够的能力或方法来应付它或有足够的“挫折容忍力”（frustration tolerance）时，挫折才能真正发挥积极作用。如果挫折的持续时间过长，超出了个体的挫折容忍力，它就会转化为内在的消极情绪状态。具体而言，组织情景中挫折有以下不良后果：

1．挫折对员工个人的影响

（1）健康损害：医学专家发现，50%～75%的疾病都与压力和挫折相关，高挫折可能导致身体上的许多疾病，包括关节炎、过敏症等，它也将潜在地引发抑郁和焦虑。不善于应付挫折的员工，工作常常没有效果，并可能伴随许多不良症状，如头疼、高血压等。当一些员工通过酗酒来解决挫折引起的身体不适时，可能进一步增加他们的健康问题。

（2）绩效倾向：挫折和绩效之间的关系就象压力与绩效之间一样，根据“耶克斯—多德森”定律，挫折犹如一根琴弦，绷得太紧或太松都不会达到理想效果，最佳程度才能获得最佳的绩效。挫折不足会导致厌倦，缺少激情，通过增加适当的挫折可使其集中精力工作。但是，如果挫折继续增加，就可能造成员工的焦虑，并消耗他们的工作精力，获得高绩效的能力就会下降。当挫折使人难以忍受时，员工可能会产生精力衰竭，无法正常工作，更谈不上绩效了。

2．挫折对组织的影响

员工由挫折引发的非理性行为反应可能会给组织造成极大的负面影响。挫折造成的身心疾病使组织的花费大量增加，另外，当员工指责组织为挫折的根源时，他们可能会采取“报复性”的攻击行为，这种行为可能是语言上的辱骂，身体上的侵害，有意的怠工及破坏公物。如故意制造工作失误，毁坏产品，制造对组织形象不利的谣言。

五、挫折管理

在管理工作中，一方面应尽量消除引起员工挫折的因素，避免员工遭到不应有的挫折；

另一方面，当员工受到挫折时，应尽量减低挫折所引起的不良影响，提高员工对挫折的容忍力。具体而言挫折管理应该注意以下几点：

（一）采取宽容的态度

对领导者来说，对受挫折者的攻击行为采取宽容的态度是很重要的。帮助受挫折者是领导者的责任之一，应耐心细致地做思想工作，避免采取针锋相对的反击措施来对付攻击行为。因为以反击对付攻击不仅是不符合互助友好原则的，而且收不到良好的效果，严重者还可能使矛盾激化。

（二）消除形成挫折的根源

1．改善组织管理制度与管理方式

如适时调整组织结构，取消有碍发挥员工积极性的不合理的管理制度，改善人事劳动制度和工资奖励制度，实行参与制、授权制、建议制等。

2．改善组织内的人际关系

组织内上级与部属间的关系不协调，过分强调单向沟通，员工没有机会向上级反映自己的意见，是影响人际关系的重要原因。因此管理者要注意改善领导与部属、管理者与被管理者的关系，建立起相互信任、相互帮助、相互支持、相互尊重的组织氛围。

（三）采用精神发泄法

由于人们处于挫折情境会以紧张的情绪反应代替理智行为，所以只有使这种紧张的情绪发泄出来，才能恢复理智状态。精神发泄法就是创设一定的情境，使受挫者可以自由地表达他们受压抑的情感。

日本一家电气公司设立所谓“情绪发泄控制室”。在这家公司的墙上挂着公司老板和蔼微笑的照片，室内放橡皮做的模拟人形，旁边架子上有各种棍子，有气的员工可以进去用棍子或拳头痛打人形靶，以发泄自己的气愤。

精神发泄还可以采用其他形式。可以让受挫者用写申诉信的办法发泄不满，当他把不满情绪都写出来时，就会心平气和了。也可以采用个别谈心的办法，或让他们在一定的会议上发表意见，领导人和同事们耐心听取他们的意见，并对其正确的方面给以充分肯定。

本章小结：

1．应激又称为压力，是人们对应激情境产生的身心反应。应激反应的模式包括警戒期、抵抗期和衰竭期三个阶段。

2．引起应激感受的客观因素称为应激源。应激源可以分为自然事件应激源和社会应激源，而社会应激源又可以分为工作应激源和生活应激源。应激感受性是个体对应激事件的敏感程度。

3．应激对个体的身体健康、心理健康和工作绩效都有影响。中等程度的应激水平最有利于工作绩效的提高，过低或过高的应激水平都会阻碍员工尽最大努力工作。

4．应激管理的策略主要有：消除应激源、远离应激源、改变应激知觉、控制应激后果和接收社会支持。员工援助计划是组织采用的一种应激管理策略，它通过专业人员对组织的诊断、建议和对员工及其家属亲人的专业指导、培训和咨询，帮助员工解决自我及其家庭成员的各种心理和行为问题，从而提高员工在企业中的工作绩效。

5．挫折是目的性行为受阻后产生的情绪，有外部性挫折和内部性挫折两种表现形式。环境因素和个人因素都会引起挫折。挫折有积极影响的一面，也有消极影响的一面。

6．人们面对挫折时可能会出现以下行为反应：攻击、退化、固执、冷漠、妥协、自我惩罚。其中妥协主要有以下几种表现方式：文饰作用、投射作用、替代作用、反向作用和表同作用等。

7．管理者应积极采取措施帮助员工应对挫折。一方面，应尽量消除引起员工挫折的环境因素，避免员工受到不应有的挫折；另一方面，当员工受到挫折时，应尽量减低挫折所引起的不良影响，提高员工对挫折的容忍力。

本章思考题：

1．确认和列出你在工作、学习或生活中遇到的部分应激物，哪个应激物最难处理？为什么？

2．举出你自己的体会（如工作上、体育活动中）来证明应激和绩效水平之间的倒 U 形关系。

3．列举几种你在日常生活中处理应激的方法和技巧。

4．描述你在日常生活中遇到的一次挫折。这次挫折产生的原因是什么？你是怎样处理的？

案例分析：

吉姆的烦恼

吉姆·布莱克驾车在拥挤的道路上缓慢地往前移动着。傍晚交通拥挤是正常的，但是今天似乎格外严重。吉姆不耐烦地点起一根烟，让他沮丧的是他和客户见面要迟到了。

作为一家出售压缩气体公司的销售代表，吉姆负担不起让客户久等的代价。市场竞争越来越激烈，其他的压缩气体供应商不停地采取各种措施抓住新的客户，而对客户来说，从一家供应商转向另一家也变得越来越容易。吉姆熄灭了还剩一半的香烟，加快油门驶向另一条路。

压缩气体的买家也知道现在市场对他们有利，因此他们不停地要求降价，同时缩短运输时间。举例来说，吉姆有一个非常苛刻的客户给他打电话要他第二天早上就把货运到很远的目的地。为了满足要求，吉姆必须填好交货表格，然后亲自确保运输队在第二天的早上将货送到。吉姆很不喜欢这样，但是没有办法。客户要求产品打折也让吉姆很不高兴，因为这会降低他的奖金。

与此同时，吉姆所在的总公司正在给销售代表施加更多的压力。他们抱怨销售代表不

够积极主动，因此要求地区总监更严格地监督每个销售代表每月的销售额。当车子再次被堵住的时候，吉姆又摸出一支烟点了起来。

两个月前，地区销售主管和吉姆谈了一次话，主要是讨论吉姆的销售额下降和丢掉了一个很重要客户的问题。尽管这还不至于被解雇，但是吉姆对工作开始焦虑，晚上也开始失眠。他需要给潜在客户多打电话，但是他发现只有在晚上回家后完成指定的文字工作后才能找到一点时间。可是这又影响到他和家人的关系。

更糟糕的是，吉姆听说德国的母公司准备卖掉美国这家分公司，也就是吉姆所在的公司。主要是竞争对手想买下这家公司扩展它的业务。吉姆担心这家公司可能不需要那么多的销售代表，那他有可能面临失业。当吉姆行驶在高速公路上望着那些红色的汽车尾灯时，头又开始痛了。

即使不丢掉工作，吉姆也担心一旦现在的公司被竞争对手收购，晋升到管理层的希望非常渺茫。尽管吉姆并不特别强烈地希望成为一名管理者，但是他的妻子非常渴望，因为这样吉姆就不用经常出差，而且薪水也不用按照每月的销售额来确定。

终于，吉姆到了客户的停车场。他迅速从他的包里找出一些阿司匹林吞下，以缓解他的头痛。他抬手看了一下手表，长叹了口气，他比约定的时间迟到了 15 分钟。

问题：

1．吉姆正在经历什么样的应激症状？

2．在这个案例中，你能区分出哪些应激源？

3．吉姆应该怎样做才能降低他的应激？

技巧库：

担忧练习

每个人都出现过担忧，这是自然的事情。然而，问题在于是积极的担忧还是破坏性的担忧。运用此技巧库可以帮助你担心一些你能控制的事情，因为担心你不能控制的事只会给你的生活增加压力。这项练习有助于你向建设性担忧目标迈进，并会增强你洞察别人担忧的能力。

1．指导语：

（1）逐条列出你最有可能担心的事情，包括你生活中各个方面的问题。

（2）将全班分成五人一组的小组。

（3）每个小组成员都要向其他成员描述他或她所担忧的每件事情。

（4）小组的其他成员一起讨论这些担忧的事情，并判断其属于下表中的哪一类？

	重要	不重要
可控	值得关注	不值得担忧
不可控	不值得担忧	不值得担忧

（5）重复这个过程，直到每个成员都将自己的担忧归类为止。

（6）活动结束后，每个小组对着全班成员报告归类结果。

2．问题讨论：

（1）其他人是怎样将你所担忧的事情归类的？绝大多数人的担忧都会归到不值得担忧的类别吗？你同意这个归类结果吗？

（2）班级成员承认自己曾经担忧过一些不重要和不可控的事情吗？

（3）你认为人们为什么会担忧一些不值得担忧的事？这样做有意义吗？这样做会有什么不利影响呢？

（4）你认为人们可以不去担忧那些不重要的事情或处于自控范围以外的事吗？

第十二章　员工卷入与管理

20 世纪 80 年代美国大陆航空公司（Continental Airlines）的服务质量相当糟糕，在业内名声很差，员工们都不敢也不愿穿着制服去别的地方，更羞于告诉别人自己在那工作。企业两次宣告破产，员工对企业缺乏认同感、满意感和忠诚度。20 世纪 90 年代中期，戈登·贝休恩（Gordon Bethune）成为美国大陆航空公司的 CEO，提出了“员工不喜欢在那工作的企业不会成功”的管理理念，并进行方方面面的管理改革。在不到三年的时间里，美国大陆航空公司无论是利润、服务还是员工满意感都得到了根本改善。企业的服务好，顾客的回头率高，企业的制服和印有企业标志的 T 恤、物品重新成为员工们的骄傲，公司效益也发生了翻天覆地的变化。

美国大陆航空公司由衰至盛的发展轨迹告诉我们，员工在心理上对企业的认同、接受及由此派生出来的积极情感（如自豪感）是一种无形的力量，可以使员工发自内心地为公司着想，为顾客服务，是企业赢利的关键所在。所谓员工卷入，就是指员工对组织事件和组织发展心理上的有效参与，表现为对组织发自内心的关心和自愿为组织服务的良好态度。它常常不能直接从员工外在的工作表现推断出来，也就是说，员工外在的工作表现并不一定意味着心理上的有效参与或卷入，员工完全可能迫于规章制度的压力、出于自身利益的考虑而表现出良好的工作行为，而内心深处并不喜欢和关心工作。有卷入地工作和无卷入地工作是两个不同层次的工作境界，也将导致不同的工作绩效和组织的不同发展结果。本章拟对与员工卷入有关的几种心理现象加以解析，并探讨它们在组织中的表现及对组织行为的影响。其中，心理契约和工作价值观可以视为员工卷入的心理前提和条件，而组织承诺和工作满意度可以视为员工卷入的心理后果和表现。也就是说，组织可以通过对员工心理契约和工作价值观的培养、教育和管理，达到员工较高的组织承诺和工作满意度，从而实现员工有效的心理参与和卷入。

第一节　心理契约

每位员工在加入企业的时候，都会签订一份劳动合同，形成双方在经济上的契约关系。个人得到一份工作、一份薪水，而企业则得到个人的剩余价值。然而，企业与个人的关系绝非如此简单，除了明文规定的书面协约之外，双方彼此还抱有一系列微妙而含蓄的期望，心理学将这一期望称为心理契约。这就好像婚姻关系，结婚证只是约束双方合法关系的“婚姻合同”，而双方能否相亲相爱、白头偕老还取决于他们彼此之间的心理契约。

一、心理契约的概念及特点

追溯其研究渊源，首先将心理契约作为一个心理学概念提出并进行详细讨论的是组织心理学家阿吉里斯（Argyris,1960），他用心理契约说明员工与雇主之间的关系。随后，这一问题被广为关注并引发了一系列研究。多数研究者（包括谢恩）都认为心理契约是组织与其成员对于双方责任与义务的期望，它有个体期望和组织期望两个水平。但是因为组织在作为契约中的一方时，其主体性并不明确，所以后来的研究多倾向于从个体角度出发，而将组织作为心理契约形成的背景来看待。心理契约的研究在 20 世纪 80 年代中期以后成为组织心理学研究的一个热点。

对于如何界定心理契约这一概念，存在不同的观点。其中比较有代表性的是谢恩（Schein,1980）的定义：心理契约是“在组织中每个成员和不同的管理者，以及其他人之间，在任何时候都存在的没有明文规定的一整套期望。”有的期望比较明确，如企业员工对薪水的期望；而有的期望则可能不明确，如对长期晋升的期望等。总地看来，相对于传统的工作合同而言，心理契约具有以下几个特点：

1．主观隐蔽性

心理契约大多是非正式的，并且是隐含的，而不是公开的，因此，它本质上是主观的。心理契约没有形成正式的文字记录，而是以心理期望的方式埋藏在契约双方的内心深处，期望着对方去理解、估测。由于这种心理契约是一种主观感觉，个体对于他与组织之间的相互关系有自己的体验与见解，因而往往会造成自己的期望与组织的理解不一致。

2．不确定性

正式雇佣契约的内容、职责、权利都是明确稳定的，不能随契约一方的主观意愿改变而改变。例如，供电公司与客户间的契约是非常正式具体的，公司必须保证不断供电，同时客户定期支付电费。而心理契约的本质是一种心理期望，它会随着工作的社会环境以及个人心态的变化而发生变化。人们在一个组织中工作的时间越长，心理契约所涵盖的范围就越广，在雇员和组织之间的关系上，相互期望和义务的隐含内容就越多，这也使心理契约的内容具有更大的不确定性。

3．双向性

心理契约是组织与组织成员之间建立的一种双向性的联系，一方面是员工对自己在组织中的权利、发展等方面的期望，另一方面是指组织对于员工忠诚、责任等方面的期望。可以说，组织与组织成员在心理契约中都处于主体地位，是完全平等的。因此，组织与组织成员在向对方提出期望和要求的同时，应多注意双向沟通，尽量去领会并满足对方对自己的期望。只有通过契约双方的相互交流、相互沟通，对组织与个人的发展达成共识，营造良好的工作环境，才能发挥心理契约的激励作用。

4．动态性

心理契约的主观性与不确定性，决定了它具有动态发展的特点，这就要求心理契约双方根据环境变化和企业发展来确定心理契约的内涵。心理契约没有固定的模式与统一的标准，适用于一个组织的心理契约不一定适用于另一个组织。例如，许多过去在心理契约中

非常重要的因素，如员工以忠诚、遵从和努力为条件换来的工作稳定感，正在逐渐减弱或占据次要位置；而一些新的内容，如对灵活性、公平性、变革创新、不断尝试的要求，在心理契约中所占的比重越来越大。因此，个人和组织都需要根据组织内外环境的变化，不断调整和完善心理契约。

二、心理契约的内容

依照承载主体的不同，员工和企业之间的心理契约可分为员工心理契约和组织心理契约两大类别。在每一类别的心理契约中又都分别含有“组织对员工的责任”（简称“组织责任”或“雇主责任”）和“员工对组织的责任”（简称“员工责任”或“雇员责任”）两个方面。

1990 年，卢梭（Rousseau）对 129 名 MBA 毕业生的心理契约内容进行调查。结果发现，员工心理契约中的雇主责任包括提升、高额报酬、绩效奖励、培训、长期工作保障、职业发展、人事支持等 7 个方面；员工心理契约中的雇员责任包括加班工作、忠诚、自愿从事职责外的工作、离职前预先通知、接受内部工作调整、不帮助竞争对手、保守公司商业秘密、在公司至少工作两年等 8 个方面。

1997 年，赫里欧和曼宁（Herriot & Manning）等人对英国各地区各行业的 184 名管理者和 184 名员工的心理契约内容进行调查。结果发现，两类心理契约中共同涉及到的组织责任项目有：（1）培训，（2）公正，（3）关怀，（4）协商，（5）信任，（6）友善，（7）理解，（8）安全，（9）有恒一致，（10）薪资，（11）福利，（12）工作稳定；员工责任项目有：（1）守时，（2）务业，（3）诚实，（4）忠诚，（5）爱护资产，（6）体现组织形象，（7）互助。

波特·马金等人详细分析了心理契约和经济契约的区别，并通过表 12.1 和表 12.2 来表示。

表 12.1　心理契约与经济契约内容和过程的区别

契 约 类 别	内　　容	过　　程
经济契约	金钱	违纪时的程序
心理契约	其他效果的奖励	有协商的权利

表 12.1 表示的是两种契约在内容和过程上的区别。“内容”说明的是交换什么，“过程”则说明怎样交换。该表只是为了说明两类契约的相对差别，并非将二者绝对对立起来。例如，任何工作契约最明显、最核心的特征就是金钱与工作的交换，但在契约中也有其他内容，还会包括休假和生病时付酬等项目。在过程上，多数工作契约都有对违反纪律和对公司抱怨行为的处理程序。但总的来说，经济契约更强调内容，心理契约则更强调过程。此外，心理契约更强调的是个人与组织间的心理关系，而不仅仅是相互之间的“交换”。人们期望被对待的方式，以及他们实际被对待的方式，对他们的工作行为有很大的影响。正如谢恩指出的，“虽然它并没有写明，心理契约却是组织中行为的强有力的决定因素。”

不论是经济上还是心理上的契约，期望和义务都是双向的。如同个人对组织的期望一样，组织也对个人有一定的期望。表 12.2 列出了两类契约类型在期望来源上的差异。

表 12.2 心理契约与经济契约的区别

契 约 类 别	个 人 期 望	组 织 期 望
经济契约	金钱	工作
心理契约	体谅	品德

一般地说，心理契约所包含的大部分内容，都可以体现在“体谅”与“品德”这两大范畴内。在“体谅”之下，个体期望在工作活动中的任何变革执行之前，他们有协商的权利，例如，在重新安排工作，引入灵活性的工作时间，或提出“不许吸烟”的政策时，都应该有商量的余地。作为对这种体谅的回报，他们可以表现出更多的有利于组织要求的品德行为，包括如果工作负荷突然增加，他们有做额外工作的愿望，或在与别人谈论其组织时，更多地谈到它的优点，支持它的目标。

需要指出的是，上述研究只是众多研究中的几项，并不能代表所有的研究。心理契约作为“一整套期望”是一个复杂的心理结构，针对其内容进行的研究因为研究者对心理契约的认识以及研究的背景不同，所得到的结果也不尽相同。而且，众多的研究还表明，心理契约的内容会因年龄、性别、工作年限、组织规模以及组织性质的不同而有所差异。

三、心理契约的类型

一些研究者根据绩效要求和时间结构两个维度将心理契约划分为四种类型。时间结构维度指的是雇佣关系的持久性程度；绩效要求维度指的是作为雇佣条件的绩效描述的清楚程度。根据两个维度划分的心理契约类型如图 12.1。

时间结构	明确界定的	没有明确界定的
短时的	**交易型** 特点：低工作模糊性 高流动率 低员工承诺 低身份感 例：销售旺季时临时雇佣的销售人员	**变动型** 特点：高工作不确定性 高不稳定性 高流动率 例：处于组织减员或公司并购过程中的员工
长时的	**平衡型** 特点：高员工承诺 高身份感 不断开发 相互支持 动态性 例：高参与性工作团队中的成员	**关系型** 特点：高员工承诺 高感性投入 高身份感 稳定性 例：家族企业中的成员

绩效要求

图 12.1 心理契约的类型

有研究者对美国一所大学的170名博士生和他们的46名导师之间合作关系的契约性质进行调查，结果证实了交易型、关系型、平衡型和变动型4种契约类型的存在。研究发现，超过75%的学生和导师都报告说他们之间的关系是“关系型”和“平衡型”。

当员工感到与组织的心理契约被违背时就会产生许多消极情感和行为，员工会认为自己受到了不公正的对待，它促使个体重新评价自己与组织的关系，并对组织承诺、工作投入、工作绩效、工作满意度和员工流动率均有不良影响。因此，管理者要根据员工心理契约的不同类型进行有针对性的管理。

具有交易型和变动型心理契约的员工非常容易流失。交易型的员工，如临时雇佣人员，具有工作任务明确，流动率高，对组织承诺和认同感低的特点，在组织中应尽量少设相应的职位，可能的情况下，相关工作任务宜进行外包处理。这类员工的心理契约中，对组织责任而言，短期报酬及承诺的时效要求较高，而对长期的福利关注不多，企业要有针对性地制定薪酬政策，实行动态管理。临时聘用人员可以根据工作表现而转变成长期合同甚至固定合同人员，提高他们对组织的认同感。

变动型的员工，如处于组织减员或公司并购过程中的员工具有较高的工作不确定性，较高的不稳定性和较高的流动率。因此当组织在进行并购或组织结构调整时，应对其中比较优秀的员工尽快明确工作任务，或者尽快签订劳动合同。

平衡型的员工，如骨干团队中的成员具有较高的组织承诺和组织认同感，能够不断自我开发，协作感也比较强，在组织中工作的时间较长。关系型的员工，如家族企业中的成员或企业高层管理者除具有平衡型的特点外，还具有强烈的身份感和稳定性强的特点，在组织中工作的时间很长。

具有平衡型和关系型特征的员工是企业的核心员工，并且在企业工作的时间较长，对企业的商业模式、客户资源等都比较了解。当其心理契约被违背后，他们对组织的情感投入减少，交易型成份加强，严重的甚至会终止自己的工作，而这些员工一旦流失，导致的人力资源重置成本是非常高的，对企业的伤害也非常大。

阅读材料：

爱立信的新玩法——心理契约

王鑫是爱立信的一名员工，曾经在7个不同部门任职，向11位上司汇报过工作。一般说来，员工在不同部门间频繁调换岗位，通常会被认为工作能力低下或者不能适合群体性工作，离职也就成为自然而然的结果。但爱立信却并非如此。

王鑫在爱立信中国公司工作了10年之后，多个部门和任职角色的变化，不仅使他没有得到负面的评价，反而因其卓越的业绩而广受尊重。最初加入爱立信时，王鑫负责产品管理，后来又担任项目顾问，现在的职务是系统方案部高级总监。他说：“技术是我的专长，我的性格也适合专项的技术开发性工作。”王鑫的同事对其也有类似的评价：沉稳而平和，适合做技术。

“但是我并不拒绝变化。”王鑫加强了语气，“过去的7次职位变动，基本上都是我主动选择的，并且每发生一次职位变化，我都会多些收获。”

王鑫认为，做产品管理工作时，要经常与相关的研发、测试、市场等部门反复沟通，使他作为管理者沟通时更加自信；项目顾问的角色，使他今天在担任技术专家时能更多地站在客户需求的角度进行功能定义。“这种变化，最终对我的职业能力有益，也对公司有益。”他说。

事实上，王鑫作为一个职员个体，从微观角度上体现了爱立信对员工的素质要求：积极主动性，把自己的发展与公司业务相结合，不断创新；具备客户服务意识，把客户的需求负责到底。

“很多爱立信员工的素质提升是在职位变动中实现的。” 爱立信中国公司副总裁马晋红认为，爱立信公司员工的内部变动，是员工和公司之间相互选择的过程。在选择中，员工获得了从事不同职务的机会，他们可以在专业化背景的基础上，增加多样化的技能，员工自身的就业安全感和对业务的参与程度也相应自然提高。

“不仅是我，在爱立信的很多同事都会在 15 个月左右调换一次岗位”王鑫说。爱立信内部岗位轮换显然已经成为一种传统，这种传统更深层次体现的是，爱立信与员工之间在变化中形成一种长期合作的心理契约关系。

哈佛大学心理学教授谢恩指出，虽然并没有写明，心理契约确实是组织行为的强有力的决定因素。尽管心理契约是学术性的专用名词，但实践表明，不管是个人职位变动，爱立信内部的组织变动，还是爱立信的方向在顺境和逆境中发生转变时，心理契约都是为公司整体带来积极动力的核心。

（来源：http://www.moeedu.com.cn/men/show.asp?id=10662&bigclassname=人力资源&smallclassname=案例专栏）

四、心理契约的影响

无论是对于个体的行为还是对于组织的绩效而言，心理契约都是一个重要的调节因素。心理契约可以作为正式文本契约的补充，降低不确定性，减少不安全感，对组织成员的工作满意度、离职倾向、缺勤率、组织承诺等具有重要影响。契约双方的一致性越高，上述几个指标的情况越好，从而组织的效果越好。但如果心理契约遭到破坏（基于对情景的知觉与主观解释），就会带来相反的结果。一般来说，当组织的成员感到组织实际履行的义务与被期望执行的义务之间出现差异时，违约现象就发生了。心理契约的违背会导致强烈的情绪反应和被背叛的感觉，程度较轻的违背会带来高离职率、低工作满意度、低信任度等后果，并将导致契约的交易性增强，即组织成员更多地关注金钱与物质方面的眼前利益。有研究指出，这种交易型关系的进一步破坏可能会导致三种行为结果：（1）公开谈判，（2）调整工作投入，（3）辞职。无论哪一种结果对组织都是不利的。值得注意的是，许多研究都已经表明，心理契约的违背现象相当普遍，最高可达 50%以上。

五、心理契约的有效管理

从组织管理的角度来看，怎样强调心理契约的重要性都不过分，因为它是影响组织凝聚力的至关重要的因素。对心理契约的有效管理一般应从建立、诊断、使用与调整四个环节着手。

（1）建立：组织与员工签订雇佣合同时，在心理契约方面达成一致并为对方所知晓和认可。要建立满意的心理契约，应注意规范选拔过程。在个体与组织最初接触的时候，双方都以他们所期望的观念来处理彼此的信息并初步建立心理契约。对组织来说，应当实事求是，而不能过分吹嘘，避免使人产生一种不切实际的期望，否则就会导致心理契约的违背。

（2）诊断：组织通过一系列的手段与方法，对心理契约的实现与变化情况进行监督与控制。及时地沟通、将心理契约公开化是最为有效的手段。

（3）使用：组织围绕已经建立的心理契约开展各项人力资源管理实践，以提高组织绩效。

（4）调整：当组织诊断出心理契约发生变化时采取有效措施。在组织发展过程中必然会伴有心理契约的变化，如果不能根据变化的情况进行及时修订与调整，就会在某个层次甚至各个层次引起不满，使整个组织陷于不利境地。

第二节　工作价值观

工作占据了人们大部分的时间，甚至成为大多数人生活中最主要的内容。那么，人们究竟是为了什么而工作的呢？在工作中最看重的又是什么呢？这就涉及人们的工作价值观。工作价值观不仅影响个人行为，还影响群体行为和整个组织行为，进而影响组织的绩效与发展。

一、工作价值观的概念

工作价值观（work values）是个人价值系统的一部分，其涵义是从价值观的涵义延伸而来的，与生活价值观密切相关。工作价值观可以说是一般价值概念的特殊用法。

工作价值观的概念最早是由赛普尔（Super）提出来的。他在查阅工作满意度的文献时发现，工作满意度与工作的某些特质如收入、成长、独立性等密切相关，后来其他人也提出了相同的看法。1957 年赛普尔将早期观察到的工作特质进行归类，并将这些因素命名为工作价值观。随后学者们提出了许多工作价值观的定义，但至今学术界尚没有一个统一的定义。

不同学者的定义大致可以分为两种取向：一种倾向于从需求满足的角度来定义工作价值观。例如，赛普尔将工作价值观定义为与工作相关的目标表达，反映了个人的内在需求及其从事活动时所追求的工作特质。另一种倾向于从信念、标准、偏好等心理特征的角度下定义。艾利泽（Elizur）认为，工作价值观即个体关于工作行为及其在工作环境中获得的某种结果的价值判断，是一种直接影响行为的内在思想体系。

综观上述两种取向，可以认为工作价值观是指员工关于自己能够从工作中获得什么成果及在工作上应该如何表现的信念。它既反映了员工想要通过工作达到的外在目标，又反映了员工想要通过工作实现的内在价值。工作价值观是整个人生观、价值观的重要组成部分，在个性心理结构中居于更加核心和基础的地位，是一种较为本质的、稳定的工作态度，因而对人的影响也更加持久而广泛。研究表明，人们的工作价值观通常是在参加工作以前的社会化过程中逐渐形成的，它将在今后工作的几十年时间内影响人们的工作动机、工作责任心、对业务的钻研精神和对职业的忠诚度。因此，工作价值观问题引起了国内外学者的广泛注意，目前国际上已经成立了工作价值观研究学会，并在美国、德国、以色列、加拿大和土耳其等国家举行了多次国际学术会议，对各个国家人们的工作价值观进行分析和比较。

二、工作价值观的结构维度

不同的研究者对工作价值观的结构维度有不同的看法。下面简单介绍几种有代表性的观点。

罗基奇（Rokeach）认为，价值观包括两种：终极价值观和工具性价值观。终极价值观是个体努力追求的最终存在状态，例如世界和平与美好、平等、智慧和舒适的生活。根据赋予的权限不同，可以将终极价值观分为以自我为中心的和以社会为中心的两种。工具性价值观是指那些有助于实现终极价值目标的行为模式，包括以道德为中心的和以能力为中心的两种。以道德为中心的工具性价值观，包括行为模式中诸如诚实和责任感之类的内容；以能力为中心的工具性价值观，则主要包括行为模式中诸如符合逻辑和自我控制之类的内容。通常，管理者更为关注工具性价值观，因为它能塑造员工的行为，使其与组织价值观相一致。

赛普尔将工作价值观划分为三大类：一是内在工作价值，即工作本身的性质和内容带来的内在满足和价值感；二是外在工作价值，即工作结果给他人或社会带来的价值；三是外在报酬，即工作给自己带来的外部回报。赛普尔根据自己的理论建立了“工作价值问卷”（WVI），包括 15 个因子。宁维卫（1996）对 WVI 进行了修订，制定了 WVI 中国版，通过因素分析得到不同人群的五因素（进取心、经济价值、社会贡献、声望、人际关系）工作价值观结构。

最常用的工作价值观结构模型是把工作价值观分为内在工作价值观与外在工作价值观两大因素（参见表 12.3）。内在价值观是和工作本身特质相关的工作价值观；外在价值观是

和工作结果有关的工作价值观。

表 12.3　内在与外在工作价值观比较

内在工作价值观	外在工作价值观
有趣的工作	薪水
具有挑战性的工作	工作保障
学习新事物	工作福利
重大贡献	社会地位
工作潜力完全发挥	与社会接触
责任感与自主能力	家庭时间
发挥创造力	休闲时间

那么在现实情况下，员工的工作价值观究竟是怎样的呢？下面的阅读材料就是一项关于我国员工工作价值观调查的结果，从中可以看出，中国员工在工作中最看重的是内在价值观，而不是薪水、福利等外在价值观。

阅读材料：

你为“什么”而工作

2005 年 3—4 月，中国人力资源开发网与新浪文化、雅虎财经等媒体合作，通过在线调查的方式，开展了一项题为《2005 年——你为“什么”而工作》的调查，共收到 7632 份有效问卷。

调查结果显示，“被人尊重”得到了 78.3%公众的认同，成为最被看重的工作价值观。此外，分别有超过七成受访者表示希望“自己在工作方面有影响力”（76.6%）、“工作对个人成长有帮助”（74.4%）、“工作上有成就感”（73.1%）和“工作能够得到认可”（72.9%）。

调查发现，女性、年轻人、普通员工以及在国有企业工作的人，在对工作价值的判断上更注重“自己的工作能否得到认可”；而男性、年长者、管理者以及在民营企业或外企工作的人则更关注自己“对工作是否有足够的影响力”。此外，年轻人更看重“工作对个人成长是否有帮助”；年长者则对“工作上的成就感”更为重视；而政府机关或事业单位的人则希望“工作稳定”。

调查中，研究人员采用聚类分析的方法，根据工作价值观的不同把员工分成六类：工作满足型、理想主义型、安逸享乐型、随波逐流型、回报驱动型以及创业型。

分析发现，30.3%的被调查者属于“理想主义者”，他们对工作条件、工作活动、工作回报以及工作上的尊重与影响力都非常注重；22.4%的人属于“创业型”，他们对工作条件少有挑剔，希望从事有意义和有成就感的工作，能有足够的自由与影响力来把工作做好；22.1%的人属于“回报驱动型”，他们看重工作所带来的回报，这种回报包括物质上、精神上等各个方面。

有 33.3%的女性属于“理想型”，而男性中这一比例只有 27.3%；相反，“创业型”

的男性占 25.9%，女性只有 19.1%。此外，“安逸型”女性的比例也超过男性，这表明女性更希望工作稳定，不需要加班加点，工作时间安排能符合她们的个人需要。

比起年长者（19.8%），更多的年轻人（24.3%）属于“创业型”。而学历越高者，越注重“创业”与“回报”，学历越低的人越看重“安逸享受”，越容易“随波逐流”。

在外商独资企业工作的人中，“理想主义者”（35.5%）的比例是最高的；民营企业中，“工作满足型”（8.5%）和“创业型”（26.4%）的人数比例最高；而事业单位或政府机关中，“安逸享受型”（17.8%）和“随波逐流型”（8.8%）的人相对较多。另外，还分别有 22.7%的国有企业职工和 22.4%的政府机关或事业单位职工属于“回报驱动型”。

从这一调查结果可以发现，中国人在工作中最看重的并不是“薪资”、“福利”等外部回报，也不是“工作稳定”、“工作环境”等外界条件，而是一些内在的回报。

（来源：http://zqb.cyol.com/gb/zqb/2005-05/16/content_4347.htm）

三、激发员工的内在价值观

工作在人们生活中占据的重要位置，正如奥地利享有崇高声誉的心理分析专家威廉•赖克所说：“爱工作和知识是我们的幸福之源，也是支配我们生活的力量。”因为工作，人们与社会的连接更为紧密，有了更真实的存在感和满足感，可以为自己一步步地超越自我而喜悦。所以，幸福并不是虚无缥缈的存在，它就存在于每日的工作之中。

激发员工的内在工作价值观实际上就是发掘工作的价值和意义。现在很多企业都通过各种方式激发员工内在的工作价值观，帮助员工从工作本身中获得幸福。世界著名企业松下电器公司，自从 1918 年 3 月成立后，在一段时期内取得了比较大的发展。但是到 1932 年年初，作为创业者的松下幸之助发现，公司再像原来那样发展已经非常危险。他认为：“我们如此拼命地工作，若没有一个意义存在是不行的。所以，不光是以过去世间普遍接受的常识来工作，还必须清楚地认识公司所肩负的使命。”因此，1932 年 5 月 5 日，松下先生召集干部开会，明确了公司的价值观：“我们的努力是为了提高全人类的生活水准。也就是说，并不单是为了公司的业绩提高，或是保证从业人员的薪资，提高他们的生活水平，而是具有更大的意义——为整个社会的繁荣作出最大的贡献。有了这样的使命感，才可以看得出我们工作的价值。”他将这一天设立为“创立日”，意指从这一天开始一个全新的企业诞生了。

意想不到的是，此举产生了惊人的效果。员工们都说：“原来只知道好好干，而不知道为何而干。而现在，真正了解到工作的意义，应该更加努力。”此后的松下电器公司有了迅速的成长。两年后，员工的数目由 1000 余名增加到 2000 名，5 年后增长为 4000 名。正是由于激发了员工的内在工作价值观，才使得松下的员工们有了更高涨的热情和干劲，学会了从工作中享受幸福。

第三节　组织承诺

我国从计划经济向市场经济的转轨打破了对工作单位转换的限制。最近的一份调查显示，大约有三分之二的劳动力在其职业生涯中至少在两个单位工作过，三分之一的人至少在三个单位工作过。年龄越低，流动的频率越高。劳动力的自由流动，一方面使组织之间的人才争夺愈演愈烈，另一方面也使员工与组织之间的关系发生了深刻的变化。在这样的形势下，对员工的组织承诺进行探讨和研究就显得日趋紧迫和重要。

一、组织承诺的概念

组织承诺（organizational commitment），也有译为“组织归属感”、“组织忠诚”等。这一概念最早是贝克（Becker）在 1960 年提出的，他将承诺定义为由单方投入产生的维持“活动一致性”的倾向。在组织中，这种单方投入可以指一切有价值的东西，如福利、精力、已经掌握的只能用于特定组织的技能等。他认为组织承诺是员工随着其对组织的单方投入的增加而不得不继续留在该组织的一种心理现象。因此，组织承诺与本章第一节介绍的心理契约是不同的，组织承诺强调的是员工对组织单方面的投入，而心理契约强调的是组织与成员的双向关系。

具体来说，组织承诺是指员工随着对组织单方面的投入增加，而产生的一种心甘情愿地参与组织各种活动的情感，是一个人对组织的认同和投入程度。组织承诺表现为：（1）希望加入某个组织；（2）愿意保留某个组织的成员资格；（3）信仰组织的价值观并接受组织目标；（4）愿意为组织的利益做出自己的贡献。

组织承诺的影响因素包括组织因素和个人因素两类。组织因素包括岗位认同、组织发展前景、人际关系等。个人因素包括性别、年龄、学历、专业知识及个性特征等。以往的研究表明，组织因素如岗位认同、组织发展前景和人际关系、福利待遇、个人在组织内的发展前景、晋升机会等与组织承诺呈正相关；在个人因素方面，女性员工的组织承诺高于男性员工，年龄的大小与组织承诺正相关，学历和专业知识与组织承诺呈负相关。

二、组织承诺的类型

目前，研究者们已经区分出了三种不同形式的组织承诺，即持续承诺、情感承诺和规范承诺。

（一）持续承诺

现实生活中经常有人仅仅因为烦于寻找新的工作而一直从事同一份工作。持续承诺（continuance commitment）就是指这种由于个人认为离开某一组织付出的代价太大而在这

一组织内供职的意愿的强度。

人们在一个组织中供职的时间越长，他们在此付出的就越多，而一旦他们离开，其损失也就越多（如退休金、亲密的友谊等），许多人仅仅因为不愿意失去这些而宁愿保留原来的工作。一般地说，这些人具有高度的持续承诺。然而，有迹象表明，当今员工的持续承诺水平已经不像以往那么高了。传统上人们找到一份工作以后便终生为之努力，许多人终生从事同一份工作，从起点开始到终点结束，这种情形在今天已经很难见到了。

2．情感承诺

情感承诺（affective commitment）是指员工由于认同组织潜在的目标和价值而持续为一个组织工作的强烈愿望。情感承诺水平高的员工留在组织中是因为认同组织的一切，并愿意协助组织达到目标。

当组织面临变革时，员工会对他们的个人价值是否仍能与他们为之工作的组织价值保持一致产生怀疑。出现这样的情况时，他们可能会问自己是否仍然属于这个组织，如果他们认为自己不再属于变革后的组织的话，就会选择辞职。

3．规范承诺

规范承诺（normative commitment）反映的是员工对继续留在组织的义务感，它是员工由于长期受组织影响形成的社会责任而愿意留在组织内的承诺。规范承诺水平较高的员工非常关心如果自己一旦离职，其他员工会有什么想法，他们不愿意让老板失望，担心同事可能会由于他们的离职而看不起他们。

阅读材料：

让员工快乐的企业

2005 年中央电视台举办了首届“CCTV 中国年度雇主调查”活动。海尔等十家员工“快乐工作”指数高的企业被评为最佳雇主，海尔集团的网上投票排名第一。在晚会现场观众及专家评审团的投票结果中，海尔集团获得了“2005 CCTV 中国年度雇主调查特别奖”。

这次调查活动的专业调查机构于 2005 年 11 月—12 月期间进入 20 家入围企业，进行企业内部员工调查，以获得员工的“快乐工作”指数。“快乐工作”指数由员工对“成就感”、“成长感”和“归属感”的感受三项要素构成。员工内部调查的结果产生了上述十家中国年度雇主名单。“最佳雇主”的概念体现了一个企业整体人力资源管理的水平，最佳雇主品牌的打造是企业人才竞争力的体现，也是持续性增长、保持竞争力的最重要的策略。最佳雇主企业往往是择业者的最佳选择。

一向以管理严格著称的海尔，为何能够成为员工心目中的最佳雇主呢？颁奖现场，海尔集团副总裁周云杰巧用中央电视台三个著名电视栏目的名称，道出了其中端倪。

第一是“绝对挑战”。海尔认为，现在缺的不是人才，而是出人才的机制。因此，作为海尔的管理者，其责任就是要通过搭建“赛马场”，为每一个员工营造创新的空间。海尔的赛马机制包含三条原则：（1）公平竞争，任人唯贤；（2）职适其能，人尽其才；（3）合理流动，动态管理。在用工制度上，实行一套优秀员工、合格员工、试用员工“三工并存，动态转换”的机制。在干部管理上，海尔对中层干部分类考核，每一位干部的职位都不是

固定的，届满轮换。海尔人力资源开发与管理的要义是，充分发挥每个人的潜在能力，让每个人每天都能感到来自企业内部和市场的竞争压力，又能将压力转换成竞争的动力。

第二是“星光大道”。周云杰说：“海尔，给每个人一个创新的空间、发展的舞台，你能翻多大的跟斗，就给你搭多大的舞台。只要你创新，人人都能够成为企业的明星、市场的明星。”

在海尔，最让人感动的是，很多在平凡工作岗位上的普通员工，能够用心去做自己的工作。一些生产线上普通的工人为了提高生产效率，搞一个技术改革，自己回家拿出钱用自己的业余时间去做。海尔的创新明星数不胜数，许多发明创造都是用发明者的名字命名的，如“晓玲扳手”、“云燕镜子”、“启明焊枪”、“申强挂钩”、“迈克冷柜”等。海尔员工通过创新，把自己的荣誉、事业、智慧和企业紧紧地结合在一起。

第三是“品牌中国”。周云杰说：“海尔的目标是创全球化品牌，也是实现每个中国人的梦想。在海尔工作的每个员工，实现自己价值的过程也就是创世界品牌的过程。因此，海尔的每个员工工作着，也快乐着。”海尔正是通过把创世界名牌这个企业的梦想和员工自我实现需求的有机融合，使海尔人在一步步的前进中品尝到了成就感，并进一步形成了海尔人独有的使命感。

海尔之所以能够成功，就是因为其内部一致的伦理观念，每个员工都围绕着这个伦理来规范和约束自己的行为，使得员工对组织保持高度的承诺感，这种高度的组织承诺感是保证海尔从激烈竞争的市场中脱颖而出的持久力量。

三、组织承诺的意义及管理

（一）组织承诺的意义

组织承诺是员工卷入的心理条件和表现形式之一，研究表明，它与工作行为的几个关键方面密切相关。

1．组织承诺水平高的员工更能坚守岗位

员工的组织承诺水平越高，他们就越不可能辞职或缺勤，组织承诺让员工能坚守自己的工作岗位，并珍惜自己的工作时间。一项大规模的调查已经证实了这一点。这项调查对美国空军学校四年制学生的辍学率进行追踪考察，发现：军校学生在开始服役时的承诺水平越高，就越不容易辍学。这表明，承诺水平能预测未来的行为，从而印证了组织承诺作为一种与工作相关的态度的重要性。

2．组织承诺水平高的员工愿意为组织做出牺牲

除了能坚守岗位之外，组织承诺水平高的员工还愿意为组织的生存和发展做出牺牲。例如，当克莱斯勒公司处于财政危机时，公司的执行总裁李·艾柯卡（Lee Iacocca）将自己的年薪降到 1 美元，充分表现了他对公司的忠诚。现实生活中，许多员工在公司遇到危机时，表现出了与企业同生死、共患难的勇气和决心，这些都是高组织承诺的直接后果。因此，可以说组织承诺是企业最宝贵的财富，特别是在企业处于危难情况时，员工高度的

组织承诺更显得弥足珍贵，可谓“一诺千金”。

（二）组织承诺的管理

管理组织承诺是一项复杂的工程，需要长期不懈的努力。大体上说，可以从以下两个方面来考虑：

1．类型管理：即针对不同的组织承诺类型采取不同的管理措施。例如，一些企业采取“利润分享计划”，规定员工在本企业工作 10 年，就可以得到 10 万元现金奖励，这是为了让员工保持较高的持续承诺。为了提高员工的情感承诺，惠普公司提出了著名的“惠普之道”，其精髓就是关怀、尊重每个人，承认他们每个人的成就，尊重个人的尊严和价值。一个小故事充分说明了这一点：有一次，惠普公司的创始人休利特在周末到一家分厂去视察，发现该分厂的实验室备品库门上了锁。他很生气，马上拿来一柄螺栓切割机，把备品库门上的锁给剪断了。星期一早上，人们上班的时候，看到门上有一张条子，上面写着：“请勿再锁此门。谢谢。威廉。”这种敞开设备器材仓库大门的做法，传递了公司对员工们充分信任的信息。为了提高员工的规范承诺，可以设法增强员工的责任感、道义感。

2．过程管理：即针对员工组织承诺形成过程不同阶段的特点采取管理措施。例如，在形成和建立组织承诺阶段，应注意招募、选择与组织价值观一致的新员工，对员工进行有针对性的培训等。特别重要的是，招募员工时不应一味地强调应聘者的经验、经历、知识、技能，而更应该注意其工作态度。在保持和维护组织承诺阶段，应注意使组织利益和员工利益保持一致，丰富工作内容等。在组织承诺提高阶段，应注意进行感情投资，而非一味地采取物质奖励。

阅读材料：

旷工：同样的行为，不同的含义

我们非常容易理解那些对组织没有承诺的员工可能会离职和旷工，然而人们通过离职所表现出的低承诺水平可能和民族文化相关。最近一项有趣的研究检验了这一观点，该研究大规模地调查了加拿大和中国员工关于旷工的态度。

总的来说，中国管理者比加拿大管理者更关注旷工率。在中国的大多数组织中，旷工是非常令人沮丧的，这种影响如此之甚，以至于那些即使没有组织承诺的中国员工也不愿意旷工在家。中国人对由于生病而旷工也皱眉头，而加拿大人一般能够接受生病这一正当理由。这和中国的文化思想一致，中国人认为一个品行端正的员工应该能够自我控制，而由于生病而旷工被认为是缺乏自我控制的表现。

然而，对于中国人反感旷工这一普遍倾向来说，还有一个有趣的例外：与加拿大人相比，中国人更容易旷工去处理个人的家庭事务，他们认为这样的做法不足为过。这其中有两个原因：第一，与加拿大的组织不同，中国员工旷工是拿不到薪资的，这样他们就不会由于白拿薪水而引起潜在的负疚感；第二，雇主一般认为员工花费一定的时间做家务是可以接受的。

这项研究的结果强调了一个关键问题：缺乏组织承诺可能会导致员工旷工（即导致员工认可旷工的态度），但仅仅缺乏组织承诺不一定导致员工的实际旷工行为，决定旷工的实

际因素是员工所处的社会文化价值观对旷工的理解。

（来源：杰拉尔德·格林伯格，罗伯特·A.巴伦·组织行为学.范庭卫等译.南京：江苏教育出版社，2005）

第四节　工作满意度

在当今的管理心理学中，工作满意度是使用最为广泛的术语之一。随着经济全球化趋势的到来，人们比以往任何时候都更加强烈地意识到，企业不仅要追求效率和利润最大化，也要追求员工满意度的最大化，达到以利益为中心和以人为中心的双重管理目标。

一、工作满意度的概念及其影响因素

所谓工作满意度（job satisfaction），是指个体对工作的认知、情感和评价性反应。工作满意度高的员工对他们的工作有积极、正面的评价，例如有的人可能会说："我喜欢和我一起工作的同事，所以我爱我的工作。"

工作满意度有时被认为是一个单维度的概念，即个体对工作满意与否。然而，它实际上涉及到了工作的各个方面，是对工作诸方面态度的集合。例如，较为流行的工作满意感量表——工作描述指标（Job Descriptive Index A，简称 JDI）——就从工作的五个方面（报酬、提升、管理、工作本身、同事）来界定工作满意度。很显然，一名员工可能会对其中几个方面满意，而对其他方面不满意。

研究表明，员工满意度每提高 3 个百分点，企业的顾客满意度将提高 5 个百分点；员工满意度达到 80%的公司，平均利润增长要高出同行业其他公司 20%左右。影响员工工作满意度的因素如下：

1．工作环境

（1）工作空间质量：对工作场所的物理条件、企业所处地区环境的满意程度；

（2）工作作息制度：合理的上、下班时间、加班制度等；

（3）资源配备齐全度：工作必需的条件、设备及其它资源是否配备齐全、够用；

（4）福利待遇满意度：对薪资、福利、医疗和保险、假期、休假的满意程度。

2．工作群体

（1）合作和谐度：上级的信任、支持、指导，同事的相互了解和理解，以及下属领会意图、完成任务情况，得到尊重；

（2）信息开放度：信息渠道畅通，信息的传播准确效高等。

3．工作内容

（1）兴趣相关度：工作内容与性格、兴趣相吻合，符合个人职业发展目标，能最大限度地发挥个人的能力，从自己的工作中获得快乐；

（2）工作强度：个人能否承受和容忍某种工作强度，一方面取决于该工作能否满足个

人需要，带来收益和满足，另一方面取决于它是否超出了个人能承受的工作负荷量。

4．企业背景

（1）企业了解度：对企业的历史、企业文化、战略政策的理解和认同程度；

（2）组织参与感：意见和建议得到重视，参加决策，企业发展与个人发展得到统一，有成就感和归属感等；

（3）企业前景：对企业发展前景看好，充满信心。

5．个人观念（这里主要是指容易引起员工不合理的不满意的个人观念）

（1）理想主义和完美主义：对企业各方面情况的理想化期望和完美主义要求，易走极端，一旦遇到困难就变得愤世嫉俗，产生不满；

（2）消极心态：将人际关系方面的问题和工作中的困难、挫折全部归因于客观原因或他人（外归因），难于沟通，人际关系不和谐，即产生不满；

（3）狭隘主义：过于重视个人利益，一旦与个人利益有冲突，产生不满情绪。

此外，一些研究表明，年龄、工作经验和性别也会对工作满意度产生影响。年长者通常比年轻人的工作满意度要高，但有趣的是，工作满意度并非随年龄的增长而保持同样的增长速度。人们通常在 30 多岁时（当他们取得初步成功时）对自己的工作更为满意，在 40 多岁时（此时他们大多抛弃了自己的梦想）工作满意度稍有下降，然而在 50 多岁时（他们已经安于生活的现状）工作满意度再次回升。有工作经验的人比没有工作经验的新手对工作更满意。妇女和少数民族群体倾向于比男子和主流群体的成员对工作更为不满。

二、工作满意度模型

图 12.2 给出的工作满意度模型具体说明了工作满意度的产生过程和机制。

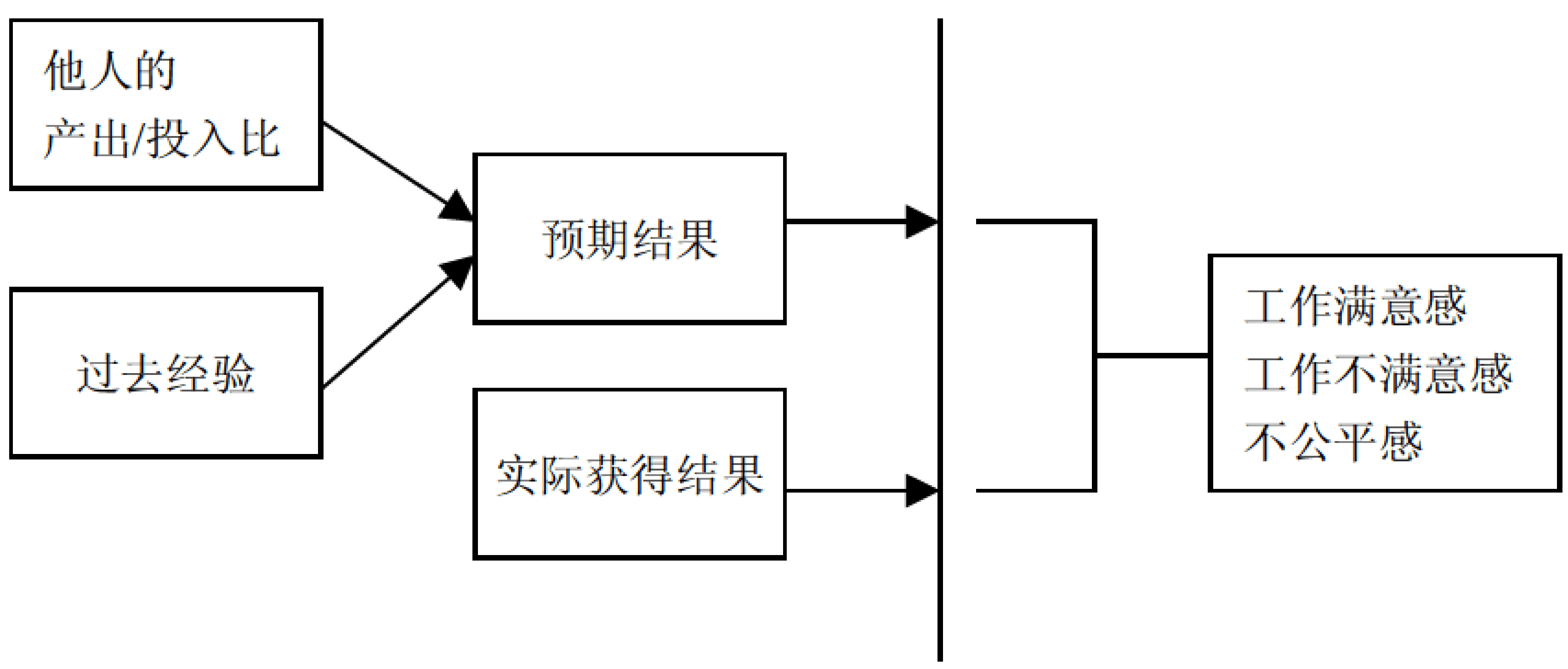


图 12.2　工作满意度模型

该模型是建立在差异理论（discrepancy theory）和公平理论（equity theory）基础上的。差异理论认为，工作满意度的大小依赖于个人实际获得的工作结果与预期结果之间的差异程度。当实际获得的结果低于预期结果时，不满意感就产生了，差异程度越大，满意度水

平越低；当实际获得的结果等于甚或高于预期结果时，工作满意感就提高了。

公平理论（参见本书第五章）认为，当个人和参照对象有相似的所得/付出比（即产出/投入比）时，才会感到公平。这和工作满意度也是相关的，因为人们对工作结果的预期在一定程度上是建立在与他人比较的基础上的。公平理论解释了为什么当人们实际获得的结果远远超过预期时，工作满意感并没有持续提高——当人们实际获得的报酬远远高于预期时，他们通常感到内疚，认为组织对待他人是不公平的。在这种过度奖赏情况下，人们首先会调高自己的预期来获得公平感受。如果过度奖赏持续不断，以至于无法通过自我调节达到平衡，不公平感就产生了，对组织的不满意感也随之出现。

总之，差异理论和公平理论认为，当实际结果达到甚至超过预期时，工作满意度将提高；然而，如果实际报酬远远高于预期以至于产生内疚感和不公平感，工作满意度就会下降。因此，该模型暗示让员工满意的一个有效办法就是弄清楚他们想得到什么，并尽可能给予满足。

最近有一项研究很好地验证了这一工作满意度模型。该研究测量了不同员工想要从工作中得到什么（如以个人方式工作的自由度、学习机会、提高机会、薪资水平等）及其对自己的重要程度，以及他们感觉自己实现这些目标的程度，同时还测量了这些员工对工作诸方面的满意度。结果发现，员工之间差异最大的方面恰恰是他们最不满意的方面，而差异最小的方面恰恰是他们最为满意的方面；而且，员工越是看重的方面，上述结果越为明显。换句话说，员工越是相信工作的某一方面非常重要，当其通过社会比较觉得自己没有达到预期程度时，工作满意度就越低。

三、工作满意度与工作行为的关系

许多公司都在想方设法提高员工的工作满意度，有些公司（如富士施乐、北电网络、美国最大的百货零售商 Sears）甚至把主管人员的奖金和员工的满意度联系起来。正如本章开头提到的美国大陆航空公司 CEO 戈登 • 贝休恩所说的那样，只有员工工作满意度水平高的企业，才能获得成功。工作满意度确实影响着员工的组织行为，但其影响力究竟有多大呢？下面着重从两个方面进行分析。

（一）工作满意度与员工离职率

当员工对工作不满意时，他们会尽量逃避和远离工作，这种现象就叫做员工离职（employee withdrawal）。员工离职的主要形式有两种：缺勤和跳槽。员工通过逃避或放弃现有工作来表示对工作的不满。

那么，员工对工作越是不满意，他们就越可能缺勤吗？研究发现，情况并非如此。也就是说，员工不满意程度与缺勤率之间的相关程度并不很高。这表明，工作不满意可能只是影响员工缺勤率的众多因素之一。例如，即使人们真的不喜欢他们的工作，但如果他们认为工作对于维持自己的生计或完成一个重大项目有重要意义的话，他们也可能不会选择缺勤；另一方面，一些员工可能非常不喜欢他们的工作，但他们只是在工作中偷懒或消极

怠工，以表示自己的不满，而不是选择缺勤。

员工离职的另一种极端形式是跳槽。员工对工作的满意度越低，他们考虑辞职的可能性就越大，同时付诸实施的可能性也越大。但研究发现，员工不满意与跳槽意向之间有较高的相关性，而与实际的跳槽行为之间却只有中等偏下的相关性。很多员工的个人因素，如职业经验、经济状况等让其难以最终作出跳槽的决定。一些研究证明：员工的经济状况以及找到一份可以接受的新工作的可能性是影响员工跳槽的重要因素。由此，研究者们推测，失业率对于员工工作满意度及员工流动率之间的关系有一定的预测作用。在失业率低的条件下，工作满意度和员工流动率的相关程度要高一些。这是因为，当失业率低时，人们认识到他们还有许多其他的工作机会，于是当对目前工作不满意时，倾向于辞职去寻找另一份工作。相反，高失业率限制了员工选择其他工作的机会，他们宁愿保持原有的工作，尽管他们或许对这份工作很不满意。因此，失业率越高，工作满意度和员工流动率的相关性越低。

（二）工作满意度与工作绩效

很多人认为，开心的员工生产率更高，但这是真的吗？换句话说，工作满意度和个人工作绩效或组织生产率直接相关吗？研究发现，总的来说，这种相关是存在的，但相关程度并不高。为什么工作满意度和工作绩效之间的相关程度如此有限呢？这其中有多种原因。

首先，一般态度并不能很好地预测具体行为。人们都有自己独特的价值观和经历，因此面对同样水平的工作满意度，每个人的反应是不同的。同样对工作不满意，有的人可能会减少对工作的投入，有的人则可能在寻找其他工作的同时依然不减少对现在工作的投入。而且，工作绩效不仅仅依靠个人的努力，还依赖于一个人的能力和资源。

其次，当工作绩效和员工报酬联系在一起时，是工作绩效影响工作满意度，而不是相反。绩效水平高的员工由于获得更多的报酬，会比由于绩效水平低而得到较低报酬的员工满意度高。工作满意度和工作绩效之间较低的相关是因为许多组织并不按员工的绩效进行奖赏，从而使高绩效未能带来高满意度。

第三，工作满意度和工作绩效之间的相关偏低还可能是因为工作满意度后果变量的多样性。也就是说，高工作满意度的后果并不一定直接表现为良好的工作行为，还可能表现为良好的组织公民行为（organizational citizenship behaviors），如在工作职责之外做一些有利于组织的事情、乐于帮助他人、积极参与组织的公益活动、营造积极的工作氛围等，这些行为并不能直接导致工作绩效的提高。

尽管工作满意度与工作绩效之间的相关并不高，但这并不意味着工作满意度对于组织而言是无关紧要的。一方面，员工的工作满意度是组织管理的目标之一，是组织管理水平的重要体现或衡量标准；另一方面，提高和维护员工的工作满意度，让工作本身变得更为愉快而有趣，是稳定员工队伍，增强员工卷入水平，树立良好企业形象，最终实现组织长远发展目标的重要途径。因此，加强员工工作满意度的管理，在防止员工对工作产生不满情绪的同时，设法提高员对工作诸方面的满意度，是管理工作的重要内容之一。

四、提高工作满意度的方法

如何提高员工的工作满意度呢？研究者们提出了以下几点建议：

（1）让工作变得更加有趣。与那些无趣和令人烦恼的工作相比，人们对自己所喜欢的工作更满意。当然有些工作本质上就是令人厌烦的，但几乎任何工作都可以注入一定程度的趣味。实际上，企业和组织可以使用一些极富创意的技巧来让工作变得更有趣。例如，让员工在工作间隙期间玩击鼓传花的游戏；拍摄别人在工作时间里幽默的瞬间并张贴在布告栏里；组织员工开展讲笑话比赛，并在午餐休息时进行评比。这些技巧可能不会使工作本身变得更令人满意，但可以通过增加工作场所的趣味来降低员工对工作的不满。

（2）组织公正。如前所述，公平感是工作满意感的前提条件。如果员工认为组织的薪资体系和分配制度等不公平，就会对工作产生强烈不满；如果员工无参与决策的权力，培训、晋升机制和运作程序不公开、不透明，也会引起员工的不满；管理层在与员工交流互动过程中居高临下、态度粗暴，也会影响员工的满意度。因此，加强组织的分配公正、程序公正和人际互动公正建设，尽可能实现分配体系和其他组织程序的公开化、透明化，是降低员工对工作不满情绪的关键所在。

（3）让员工从事感兴趣的工作。人们都有自己的兴趣和爱好，但这些很少能够在工作中得到满足。研究表明，员工从事的工作越符合他们的个人兴趣，他们对工作就越满意。现在许多企业，如美国电话电报公司、IBM、福特汽车公司、壳牌石油公司以及柯达公司等，都在为员工提供系统的测试和咨询，以便他们的工作职位能够与其技能和兴趣相匹配。还有一些公司，如可口可乐公司、迪斯尼公司等，甚至为员工提供个人化的咨询，让他们识别自己的个人兴趣和职业兴趣，并使二者尽可能匹配。

（4）避免乏味的重复性工作。研究发现，多数从事乏味的重复性工作的员工工作满意度很低。如果允许员工在工作中通过自己的方式取得成功，那么他们对自己的工作就会更加满意。

阅读材料：

赛夫为面包店的故事

不久前，赛夫为（Safeway）在俄勒刚州克拉卡马斯的面包店出现了严重的问题。该地区的130名面包工人在工作中经常出现缺勤、离职和工作事故，他们对工作惶惶不可终日。这种局面非同小可，仅仅在一年中，工作事故导致了1740个工作日的损失，这可是一个高昂的代价。当然事故仅仅发生在员工极度怠工的时候。在生意冷清的时段中，如周末晚上，如果听到有8%的员工请病假，你根本就不应该感到奇怪。该地区的员工中几乎没有人干满一年。情况很清楚，在这个组织中员工对工作不满的反应简单而明了，那就是离职，这显然是破坏性的。

是什么使得这些员工如此不满呢？调查的结果是：管理者对待员工无礼而粗暴的态度。

该地区的分公司主管是个有着极强控制欲，而且莽撞无礼的家伙，在他手下，员工无不感到无能和沮丧。

在进行了问卷调查和访谈，弄清楚问题所在之后，管理层认识到他们必须扭转这一局面。管理层认识到问题的根源在于他们武断的管理方式，因此他们彻底改变了管理方式，放宽了对员工的限制，而代之以开诚布公的态度和赋予员工充分的自由。同时，员工可以参与解决卫生和安全问题，管理层还鼓励他们提出改进意见。

管理层的改进成果卓著：由于事故而损失的工作日从每年的1740天下降到只有2天，旷工率从8.0%下降到0.2%，员工离职率从100%下降到不足10%。很明显，赛夫为面包店在管理改进中花费了大量的心血，消除了员工对工作不满所带来的消极影响。

赛夫为公司的管理层做了些什么？首先，他们表现出了对员工明显的尊重，向员工昭示了这样的信息：他们把员工的利益时刻放在心上。同时他们将决策权下放。换句话说，管理层允许员工自己作很多决定，一旦允许员工自由地参与决策，他们的满意度就能得到提高，因为他们感到可以在组织中有所作为。相反，如果决策权集中在少数人手中，员工可能会感到无能为力，从而导致对工作的不满。

赛夫为面包房的管理方式的改变说明了工作满意可以在管理者的控制之下得到改善。这个案例也提供了一个颇有价值的技巧：尊重并允许员工参与工作决策，可以提高他们的工作满意度。

（来源：杰拉尔德·格林伯格，罗伯特·A·巴伦.组织行为学.范庭卫等译.南京：江苏教育出版社，2005）

本章小结：

1．员工卷入是指员工对组织事件和组织发展心理上的有效参与。心理契约、工作价值观可以视为员工卷入的心理条件，而组织承诺和工作满意度可以视为员工卷入的心理后果和表现。

2．心理契约是指在组织中，每个成员和不同的管理者，以及其他人之间，在任何时候都存在的没有明文规定的一整套期望。心理契约与经济契约不同，其本质是对无形的心理内容的期望。

3．心理契约大多是非正式的，并且是隐含的，而不是公开的，因此具有主观性、不确定性、双向性和动态性等特点。根据绩效要求和时间结构两个维度可以将心理契约划分为交易型、变动型、平衡型、关系型四种类型。心理契约可以作为正式文本契约的补充，对组织成员的工作满意度、离职倾向、缺勤率、组织承诺等具有重要影响。心理契约的有效管理一般应从建立、诊断、使用与调整四个环节着手。

4. 工作价值观是指员工关于自己能够从工作中获得什么成果及在工作上应该如何表现的信念。它既反映了员工想要通过工作达到的外在目标，又反映了员工想要通过工作实现的内在价值。最常用的工作价值观结构模型是把工作价值观分为内在工作价值观与外在工作价值观两大因素。激发员工的内在工作价值观实际上就是发掘工作的价值和意义，可以

通过各种方法帮助员工从工作本身获得意义和价值。

5．组织承诺是指员工心甘情愿地参与组织各种活动的情感，是一个人对组织的认同和投入程度。有三种不同形式的组织承诺，即持续承诺、情感承诺和规范承诺。一般可以从类型和过程两个方面加强员工组织承诺的管理。

6．工作满意度是指个体对工作的认知、情感和评价性反应。影响工作满意度的因素主要有工作环境、工作群体、工作内容、企业背景、个人观念等。工作满意度与员工离职率、工作绩效有一定相关，但相关程度并不太高，这其中有多方面的原因。尽管如此，提高员工的工作满意度仍然是管理工作的重要内容之一。让工作变得更加有趣，组织公正，让员工从事感兴趣的工作，避免乏味的重复性工作，均可以帮助管理者提高员工的工作满意度。

本章思考题：

1．十几年前，人们大多希望能有一份稳定的工作，组织也希望员工能忠诚于自己，但现在，员工与组织之间的心理契约似乎发生了变化。你认为现在的员工与组织之间的心理契约发生了哪些变化？将来的趋势会怎样？

2．你应一家金融服务机构的邀请去帮助他们处理严重的人员流失问题，该机构里几乎没有一个员工能在自己的工作岗位上做满四个月。这不仅使公司利益严重受损，而且迫使公司不得不花费大量资金培训新员工。你认为该公司存在哪一种类型的组织承诺问题？如何在组织内提高这种承诺的水平？

3．假定一家企业最近进行了一次员工工作态度的问卷调查，结果显示有86%的员工回答总体上对工作很满意和比较满意，但对工作的许多细节方面不满意。你认为这家企业员工的工作满意度存在问题吗？为什么？

4．有人说“为了生活而工作”，也有人说“为了工作而生活”。你同意哪一种观点？为什么？

5．一名员工对他的工作和组织非常不满意，但却没有试图跳槽去寻找一份新的工作。你如何理解他的行为？这名员工在工作上会有什么表现？

6．一名员工对工作各方面都很满意，却没有获得良好的工作绩效。为什么？

案例分析：

糟糕的石油开采公司

苏珊教授正在办公室备课，约翰轻轻地敲了几下她开着的办公室的门，介绍自己道：“您好，教授，我想您可能不认识我了。”苏珊教授带的是个大班，但是她想起来了，约翰是她几年前《管理心理学》课上的学生。约翰毕业后去了一家石油开采行业，现在把工作辞了又回到学校继续读书。

“欢迎回来！”苏珊把约翰领进了办公室。“我听说你曾在一家油田工作。怎么样？”

“还好，教授，”约翰说，“我在一家石油开采公司工作了两年。当时我特别想去这家公司。据说这家公司待遇很好而且有不少假期，因此有很多人都想来这工作。经过激烈的竞争，我很幸运地成为这家公司的一员。由于这次招聘吸引了很多人来应聘，因此引起了媒体的关注，一家报纸还报道了这件事。

所以，我们这些被选中的人都非常开心，而且感到很自豪。我对此次的招聘官们印象非常深刻，因为他们似乎特别关心我们在井上工作的安全及福利问题。后来，我才知道他们都来自一家咨询公司，而且专门负责员工招聘。实际上，在那次招聘会上这家公司的管理者一个也没出现。

进了这家公司工作后，我们感到非常震惊，我们都没想到会做那么糟糕的工作，在凛冽的寒风中我们干着最脏的活。尽管这样，在头几个月里我们大多数人还是向管理层表示我们很希望把工作做好。而且我们一起被招进来的这些人相处得很好，建立起了很紧密的联系，这可以帮助我们战胜恶劣的天气和完成艰难的任务。

可是这家公司的管理者都是令人厌恶的工头。刚开始的几个月，我们尽量忍受各种虐待。后来由于要修理机器，生产被迫停了两次，这下情况变得更糟糕了。这家公司的高层对此很不满，给基层管理者施加了更多的压力，他们又把这种压力转嫁到我们头上。

他们开始不顾安全程序一个劲地催促我们快点干活，还经常当着其他员工的面谩骂、侮辱员工。有些同事被解雇了，还有些同事辞职了。有一天我也差点被解雇，仅仅因为我的老板认为我故意偷懒。他不知道，也根本不关心我正在连接的设备坏了。有好几个同事尽量避免见到上级，尽可能少做工作。一些员工开始撒谎说背痛而请假不上钻台。

在这些令人恶心的上级控制之下，我们对钻台上的问题一无所知，这些上级让我们不要对那些与我们无关的事太感兴趣。但是钻台上的问题以及以后的合同问题，对于我们这些不想辞职的人来说非常重要，我们的工作是否有保障依赖于钻台的生产水平以及能否签到合同钻新的油井。

由于有人不正确地操作了机器使得皮特因工死亡，这事引起了很大的关注。事故调查团认为皮特的死是因为他没有经过系统的培训，并且在没有任何安全警告的情况下去工作。当调查团还在调查事故的时候，好几个工人决定在钻台上成立工会，并且很快就有很多人签名加入。这让这家公司乃至整个石油开采行业非常震惊。

从那以后，公司管理层想方设法欲解散工会，他们甚至给钻台派来一个安全检查官，可是我们发现他是公司雇来的一个顾问，目的是破坏工会组织。

教授，两年前我怀着满腔热情来到这家公司，结果在上个月非常失望地离开这家公司。这真的让我很苦恼，因为我总是被告诉不管情况多么恶劣，要尽你的全力去工作。”

问题：

1．请用工作满意度模型解释一下为什么这家公司的员工对他们的工作不满意。

2．这家公司员工表达他们工作不满意的方式有哪些？

3．在这家公司工作的两年里，约翰对这家公司的承诺感在不断地下降，直到离开公司。请问影响他组织承诺的因素是什么？

技巧库：

你的组织承诺水平如何？

1．指导语：

在下面每一个陈述的左边空白处写下最能反映你同意程度的数字（1.根本不同意；2.不同意；3．同意；4．比较同意；5．完全同意）。

__（1）我留在原来的工作岗位上，不是因为我想留而是因为我不得不留。

__（2）我强烈地感觉到我属于某个组织。

__（3）一旦我在某公司工作，我就不愿意离开。

__（4）离开我的工作，需要作出很大的个人牺牲。

__（5）我感觉到我和公司紧密相连。

__（6）如果我离开公司，老板会很失望。

__（7）我没有其他选择，只能留在原来的工作岗位上。

__（8）我感到自己是所工作的公司大家庭中的一员。

__（9）我感到有强烈的责任心留在原来的工作岗位上。

__（10）如果我放弃现在的工作，我的生活会一团糟。

__（11）在这个组织中度过我工作的剩余时间，我感到很开心。

__（12）我留在原来的工作岗位上，是因为人们会因为我的离开而看轻我。

2．记分方法：

将（1）、（4）、（7）、（10）题的得分相加，反映你持续承诺的得分；

将（2）、（5）、（8）、（11）题的得分相加，反映你情感承诺的得分；

将（3）、（6）、（9）、（12）题的得分相加，反映你规范承诺的得分。

3．思考题：

（1）这个问卷揭示你哪一方面的承诺最高？哪一方面的承诺最低？这些承诺差别很大还是很相似？

（2）这个问卷在多大程度上反映了你的组织承诺水平？在多大程度上涉及到你想放弃工作和接受一个新职位的利益问题？

（3）你对这些问题的回答和你的同事相比如何？你的反应和他们的反应一样吗？你认为这是为什么？



第五篇　行为管理

第十三章　领导概述

1972 年春天，一架载有机组成员和 40 名乘客的飞机飞越安第斯山脉，其中大部分乘客是乌拉圭业余橄榄球队成员，他们此行是去参加在智利举行的球赛。飞机没能到达它的目的地，它撞上了冰雪覆盖的山峰，机体分裂成几块滑入了陡峭的山谷，最终停止在齐腰深的雪层上。相当数量的人在撞机发生时或一天后死去。有 28 位乘客幸存下来，但是，他们的处境也非常糟糕，天气极度寒冷，食物匮乏，有些乘客在撞机中受伤。在接下来的几天里，几位乘客精神错乱，还有几位因受伤而死去，那些基本上没有受伤的乘客打算做些事情来增加他们生还的几率。

有几个人努力修正飞机残骸用来抵御严酷的天气，另一些人设法找水和食物，还有些受过医学培训的人照顾伤者。幸存者们相信他们会被发现，但是随着时间推移，他们的感觉在慢慢消失，取而代之的是绝望。几周过去了，仍看不到营救队的踪影，活下来的乘客打算派出几支探险队来寻找逃生的最佳路线。由于山上空气稀薄、积雪较深，行动非常费力，身体最健康的人被挑选了出来，远足的结果令人失望，探险队认为他们所处的位置是在安第斯山脉的中央，走出去几乎是不可能的。就在这时，一场雪崩击中了飞机残骸，又有几个人死去。

活下来的幸存者看到等待营救无望，觉得生还的唯一希望还是要靠他们自己。于是，三名身体最健康的乘客被挑选了出来，组成了最后的探险队，其余的人都致力于提高这次探险的成功几率，三名探险队员得到了最多的食物以及各种必须的供应品，在飞机失事两个月后，探险队出发了。他们在最崎岖的地形上跋涉了 10 天之后，遇到在智利放牛的农民……最后，另外的 14 名幸存者获救了。

这是一个伟大的奇迹！不同领域的专家和学者从不同的角度研究他们取得成功的因素。组织心理学家发现，在这次逃生过程中，有一位名叫帕拉多的男孩发挥了关键作用。他看上去笨拙、羞涩，此前在体育和社交场上都是“候补队员”。这位似乎不可能成为英雄的人这次所表现出来的勇气、乐观、公正和情感支持使他在所有幸存者中成为最受爱戴和尊敬的人，混乱时他给人们分工，绝望时他给以鼓舞，冲突时他能妥善处理人们不同的需要与个性，关键时刻他做出了正确的决策与选择，许多幸存者说：“没有他，我们的生还是不可能的。”

帕拉多实际上实施了领导行为，他也自然地成了这支逃生团队的领导者。那么，什么是领导和领导者？什么样的领导才是有效的领导？在本章中，将对这些问题作出回答。

第一节　领导的概念和功能

一、什么是领导

关于领导的概念，不同的学者从不同的角度或侧面做出了不同的论述，概括起来，比较具有代表性的看法有以下几种：

（一）领导是一种行为过程

泰瑞（G. B. Terry）认为："领导是影响人们自动地达成群体目标而努力的一种行为。"斯托格狄尔（R. M. Stogdill）认为："领导是对一个组织起来的团体为确立目标和实现目标所进行的活动施加影响的过程。"赫姆菲尔（J. K. Hemphil）认为："领导是指挥群体在相互作用的活动中解决共同问题的过程。"

（二）领导是一种影响力

坦南鲍姆（R. Tannenbaum）认为："领导就是在某种情况下，经过意见交流过程所实现出来的一种为了达成某个目标的影响力。"阿吉里斯（Argyris）指出："领导即有效的影响。为了施加有效的影响，领导者需要对自己的影响进行实地的了解"。理查德 L. 达夫特（Richard L. Daft）认为："领导是在领导者和追随者之间有影响力的一种关系。"

（三）领导是一种权力

杜平（R. Dupin）认为："领导即行使权威与决定。"科·杨（K. Young）认为："领导是一种统治形式，其下属或多或少地愿意接受另一个人的指挥和控制。"弗兰奇（J. French）认为："领导是一个人所具有并施加于别人的控制力。"

（四）领导是一门艺术

孔兹（H. Koontz）认为："领导是一门促使其部属充满信心，满怀热情来完成他们任务的艺术。"

从上述各种定义中，可以看出，对领导这个概念的认识颇不一致，但各位学者的表述有助于全面理解领导的概念与实质。首先，要将"领导"与"领导者"这两个概念区别开来。在英语里，"领导"（leadship）与"领导者"（leader）是两个不同的单词。在汉语里，"领导"既是名词又可作动词，通常人们习惯把领导者称之为领导，把领导者的行为也称之为领导。事实上，领导者是实施领导行为的人，而领导则是领导者实施领导行为的过程。在这里，领导行为是关键，正是领导行为造就了领导者。凡是实施了领导行为的人（即便他不是上级指派的"领导者"）都是真正意义上的领导者，凡是没有实施领导行为的人（即便他是上级指派的"领导者"）都不是真正意义上的领导者。换句话说，处于"领导者"岗位上的人的行为并非一定属于领导行为，而处于非"领导者"岗位上的人的行为也并非都

不属于领导行为。“一个新来的 CEO 充其量只是企业的‘头’，他能否变成‘领导者’取决于他的所作所为和人们对其所作所为的知觉”，这句话一语道破了领导者和领导的本质和内在联系。第二，领导行为是一个动态的过程。这个过程由三方面相关的因素所构成，即领导者、被领导者和环境。其中，领导者是起主导作用的因素，被领导者、组织环境是影响领导有效性的重要因素。领导行为是由这三个因素组成的复合函数，用公式表示为：领导行为＝f（领导者，被领导者，组织环境）。因此，研究领导行为时，必须充分考虑各种因素的作用及相互关系。第三，领导是有目的的活动。领导行为的目的是领导者指引和影响被领导者实现团体或组织的目标。

根据上述分析，可以把领导表述为：领导是在一定条件下，指引和影响个人或组织，实现某种目标的行动过程。其中，把实施指引和影响的人称为领导者，把接受指引和影响的人称为被领导者，一定的条件是指所处的环境因素。领导的本质是人与人之间的一种互动过程。

二、领导与管理

从广义上来看，管理行为是领导行为的组成部分。领导行为包含着一些管理活动，人们一般把组织中的中层和基层领导者称为管理者，其领导行为称为管理活动。从狭义上来看，二者有着本质的区别，领导不能代替管理。管理可以这样来定义：它是通过计划、组织、配备、命令和控制组织资源，从而以一种有用的、高效的方法来实现组织目标。领导和管理的区别具体体现在以下几个方面：

（1）任务不同。领导注重愿景和长远未来，要求创立一个可以激发组织和团队士气的未来愿景，并制定长期的战略规划，为实现组织愿景进行相应的变革。管理关注的是具体目标和短期结果，侧重于为获得特定的、具体的结果而制定详细的计划和日程安排，然后分配资源以完成计划。

（2）目标不同。领导者是设计师，其主要任务是解决组织中带有方向性、战略性和全局性的问题，领导过程是通过与员工交流和沟通组织的愿景，发展出共同的组织文化和一套核心的价值观，从而把组织引向渴望的未来。管理者是工程师，其主要任务是实现组织的效率和效益。管理过程是为了实现既定的目标，建立一定的管理框架，然后用员工来填充这个框架，制定相应的政策、程序和系统来指导员工并且监督计划的实施。

（3）角色差异。一般称领导者为“帅才”，即具有“运筹于帷幄之中，决胜于千里之外”的领导才能，帅才可以不必过问方案的实施细节，“能将将者，谓之帅才也”。管理者一般称为“将才”，“能领兵者谓之将才也”，将才必须重点考虑方案的实施细节。另外，领导者一般居于组织或团体的高层，具有相当的影响力，人数较少；而管理者包括组织或团体的中低层管理人员和从事业务管理职能的一般管理人员，人数相对较多，如财务、营销管理人员等。领导者与管理者特征比较如表 13-1 所示。

表 13.1 领导者与管理者特征比较

管理者特征	领导者特征
管理	创新
维持	发展
集中于系统和结构	集中于人
信赖控制	激发信任
短视的	远视的
询问如何与何时	询问什么与为什么
模仿	首创
接受地位	挑战地位
正确地做事	做正确的事

三、领导的功能

领导者在领导活动中所表现出来的行为就是领导行为。领导行为的影响和作用表现为领导功能。通常，领导的功能表现为以下三个方面。

（一）组织功能

领导的组织功能是指建立组织管理机构、科学组织生产以及达成组织目标的功能。首先，组织的有效运作客观上要求必须有清晰的职位层次与管理控制幅度、流畅的信息沟通与反馈系统、有效的协调合作系统、强有力的指挥系统与指挥中心、高效的执行机构以及严密的控制与监督系统等，特别是在现代大型组织中，还需要建立信息决策中心、事业发展中心等咨询系统，这些系统的建立和维持都需要领导者通过其领导行为和过程来实现。其次，实现组织目标是领导者的根本任务和组织功能。这其中包括对企业或组织内外部环境条件进行科学分析、制定组织的总目标、对总目标按照若干级别进行分解、按时间进程进行长期目标、中期目标、短期目标与近期目标的分解与结合、为实现组织目标合理地组织和使用人财物、组织管理系统运作程序与组织规章制度等。第三，领导的组织功能还包括根据环境条件的变化所进行的组织变革、组织发展甚至组织重构等内容。

（二）激励功能

激励功能是领导的主要功能之一。领导者各项决策的制定、组织目标的实现都是借助于发挥和调动团队和员工的积极性来完成的。任何一位领导者，如果不能有效地发挥激励功能，即使目标决策再好、组织设计再合理、管理制度再完善、管理手段再科学，也难以实现既定的目标，也不能算是一个好的领导者。领导的激励功能主要包括：（1）使用“员工参与”激励。领导者要善于将员工的个人目标与组织目标相统一，实行参与式的民主管理，鼓励员工参与目标制定和决策的积极性，创造有利于员工参与的组织和团队氛围，达

到激励员工的目的。（2）发挥“关怀”的激励作用。领导者要关心员工的生活，善于和员工进行思想交流，主动了解员工所思所想，拉近与他们的心理距离，这样员工容易产生组织公民行为，提高对组织的忠诚度，从而调动工作积极性。（3）运用“奖励”激励。组织奖励对员工具有较大的激励作用，领导者要适时地对表现优秀的员工进行奖励，一方面能对被奖励者起到激励作用，另一方面能起到引导和示范的激励作用。（4）实行“素质提高”激励。首先，领导者要鼓励员工确立个人发展目标和职业生涯规划，激发员工的成就需要和自我实现需要；另一方面，组织要努力创造条件为员工提供各种在职培训和外出培训机会，对于表现好的员工，组织可以有计划地根据员工的个人需要选派他们到高校、大企业等参加学习，提高他们的业务能力，这样既为组织的发展培养了后备人才，也能起到激励员工的效果。

（三）角色功能

领导者承担的角色功能主要有以下三类：第一类是人际关系角色，如扮演组织元首，代表组织与外界打交道以及负责组织内外联络等；第二类是情报性角色，主要扮演监听与侦探、传播者、发言人等角色；第三类是决策者的角色，主要扮演企业家、危机管理者、资源分配者以及谈判者的角色等。心理学家明兹伯格认为，领导者只有认清自己所扮演的角色，调整好角色的时间比例与时间分配，才能获得最佳的组合效果。

科列奇（D.Krech,1978）把领导者的角色功能划分为 14 种，分别是：（1）方针制定者：确定组织目标，制定方针政策；（2）计划者：规划实现目标的手段与步骤；（3）执行者：调整组织活动，监督决策的执行；（4）专家：信息情报专家、专业技术专家与组织管理专家；（5）内部关系的协调与控制者；（6）对外的代表者：代表组织同外部群体和个人交涉；（7）赏罚权限的执行与控制者；（8）组织和团体内部纠纷的调整与仲裁者；（9）具体执行的示范与榜样者；（10）组织和团体的象征，即团结统一的中心与精神支柱；（11）替身作用：替组织和团体承担责任；（12）团体规范的提倡者；（13）理想者：众望所归者；（14）奉献者：组织或团体失误时员工发泄不满的对象与攻击的靶子。

第二节　传统的领导有效性理论

自 20 世纪初以来，关于领导有效性的研究一直是管理心理学、组织行为学等领域关注的焦点。随着研究的不断深入，各种领导理论相继出现，逐步形成了关于领导有效性的特质理论、行为理论和权变理论等三大理论派别。

一、领导有效性的特质理论

领导有效性的特质理论强调领导者自身一定数量的、独特的、并且能与他人区别开来的品质与特质对领导有效性的影响。早在 20 世纪 30 年代，一些心理学家就把注意力集中

到那些在一定程度上可以成为伟人的领导者身上，希望发现领导者与非领导者在个性、社会、生理或智力因素方面的差异。通过大量的实证调查研究，特质理论的支持者找到了一些证据，证明领导者在社交性、坚持性、创造性、协调性等方面都超过了普通人，其个性特征与一般人也有区别。代表性的研究结论主要有：

管理学家、社会系统学派的代表人物切斯持·巴纳德于 1938 年在《经理人员的职能》一书中，认为领导者应该具备的基本特质是：（1）活力和忍耐力；（2）当机立断；（3）循循善诱；（4）责任心；（5）智力。

心理学家吉伯（C.A.Gibb）在 1969 年的研究报告中，指出天才的领导者具备 7 项特质：（1）善言辞；（2）外表英俊潇洒；（3）智力过人；（4）具有自信心；（5）心理健康；（6）有支配他人的倾向；（7）外向而敏感。

心理学家斯托格迪尔（R.M.Stogdill）于 1974 年在《领导手册》一书中，进一步提出了领导者应该具备的 10 项特质：（1）才智；（2）强烈的责任心和完成任务的内驱力；（3）坚持追求目标的性格；（4）大胆主动的独创精神；（5）自信心；（6）合作性；（7）乐于承担决策和行动的后果；（8）能忍受挫折；（9）社交能力和影响别人行为的能力；（10）处理事务的能力。

1971 年，心理学家爱德温·吉色利（Edwin E.Chise）在《管理才能探索》一书中，采用语义差别量表法，选择分布于交通、制造、通信、财政金融、保险、公用事业等 90 个不同组织，年龄跨度从 26～42 岁，学历层次 90%为大学程度的 306 名管理人员进行研究，得出了包含 3 大类、13 个因子的领导特质：

第一类特质为能力，包括管理能力、智力、创造力三个因子。

第二类特质为个性品质，包括自我督导、决策、成熟性、工作班子的亲和力、男性的刚强或女性的温柔等五个因子。

第三类特质为激励，包括职业成就需要、自我实现需要、行使权力需要、高度金钱奖励需要、工作安全需要等五个因子。

遗憾的是，研究者们没有找到区分有效领导者与无效领导者的特质模型。不过，研究者还是取得了一些研究成果，如发现领导者有 6 项特质不同于非领导者，即进取心、领导意愿、正直与诚实、自信、智慧和与工作相关的知识。此外，还发现高自我监控者在调节自己行为以适应不同环境方面具有很高的灵活性，比低自我监控者更易于成为群体的领导者。

特质理论的最近研究发现，与领导有效性有关的关键能力有：（1）驱力：追求目标的内在动机；（2）领导动机：使用社会化的权力影响他人以获取成功；（3）正直：可信赖性以及把话语变为行动的意愿；（4）自信：相信自己的领导才能，坚信有能力实现目标；（5）智慧：处理信息、分析选项并发现机会的能力高于一般人；（6）商业知识：了解其运作的商业环境，有助于准确决策和为组织带来成功；（7）情绪智力：基于自我监控的人格，确保优秀领导者具有更强调情境敏感性以及在必要时适应环境的能力。

二、领导有效性的行为理论

从 20 世纪 40 年代，关于领导有效性的研究发生了转变，进入了“行为”理论阶段，即从领导者的行为方式来探索成功的领导模式。代表性的研究成果有：领导行为四分图模式、管理坐标图和 PM 领导行为理论。

（一）领导四分图模式

1945 年，美国俄亥俄州立大学首先开创行为理论的研究。他们从 1000 多个领导行为特征中，不断地提炼概括，最后归纳为“关心人”和“抓工作组织”两个行为量纲。他们发现，这两种行为在不同的领导者身上所表现出来的强弱程度不尽一致，由此可以形成图 13.1 所示的“领导行为四分图”。

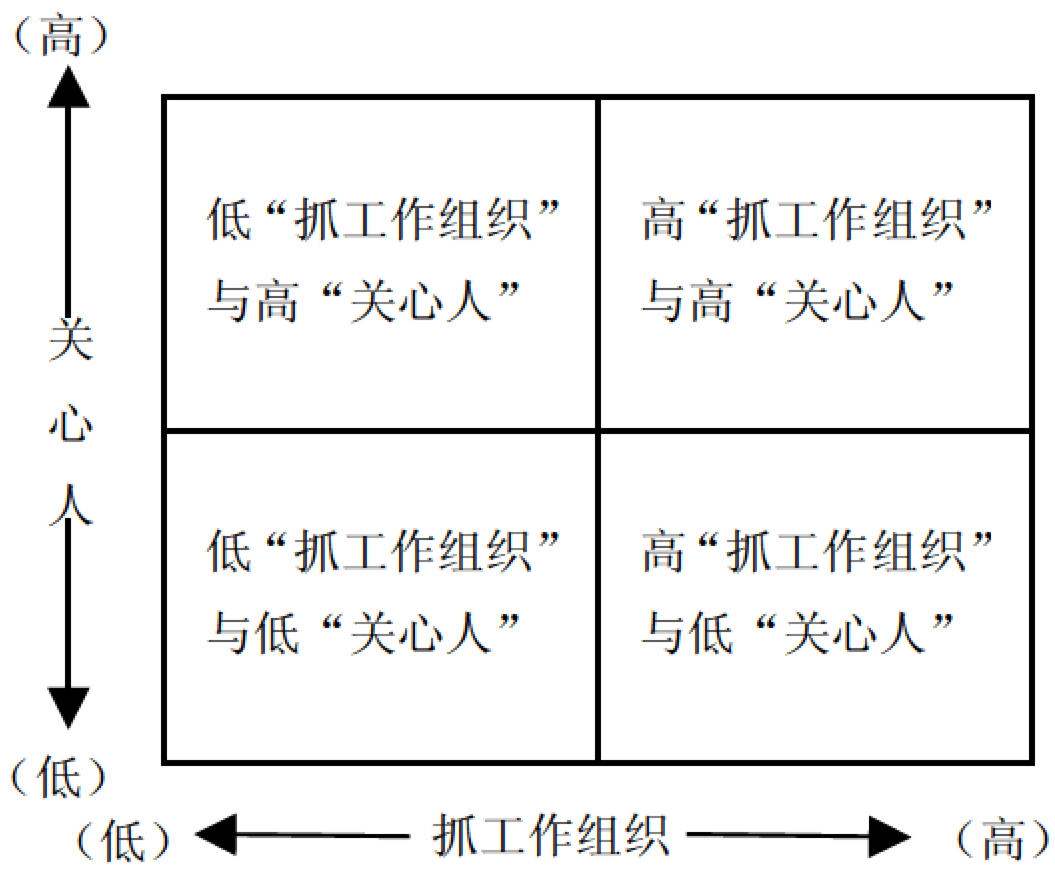


图 13.1　领导行为四分图

在领导行为四分图的基础上，密执安大学的心理学家提出了领导行为的两个维度：面向员工和面向生产。面向员工的领导者倾向于较高的集体生产，与员工较高的满意度相联系；而面向生产的领导者则倾向于较低的集体生产，与员工较低的满意度相联系。

（二）管理方格理论

管理方格理论是管理心理学家罗伯特·布莱克（Robert R. Blake）和简·莫顿(Jane S.Mouton)提出来的。该理论认为，有效的领导者应该是一位既关心工作，又关心员工的人。关心工作是指领导者更关注与工作过程相关的因素，如生产过程、产品质量、销售以及服务质量等。关心员工包括对友谊、自尊以及公平的报酬等方面的关注。

罗伯特·布莱克和简·莫顿指出，任何一位管理者的管理策略都会体现在对这两个基本要素的关注程度上。一个管理者可能对生产高度关注而对人不太关心，或者相反，也可能处于两者的中间状态。因此，这两个要素的组合可以清晰地揭示出领导者的类型及其管理方式。如果分别以这两个要素为纵坐标和横坐标，并将这两个坐标轴都划分成九个刻度，便形成了图 13.2 所示的能表示 81 种领导方式的管理方格。要评价某个管理人员，可在 9

点评分量表上对其关心员工和关心生产的程度分别进行量化记分，然后在管理方格图上找到两个分数对应的交叉点，就可以确定他所属的领导方式或管理类型。例如，某领导者关心员工的程度达到了 7 分，而关心生产的程度只得到 2 分，那么，他就是 2 •7 型的管理者。

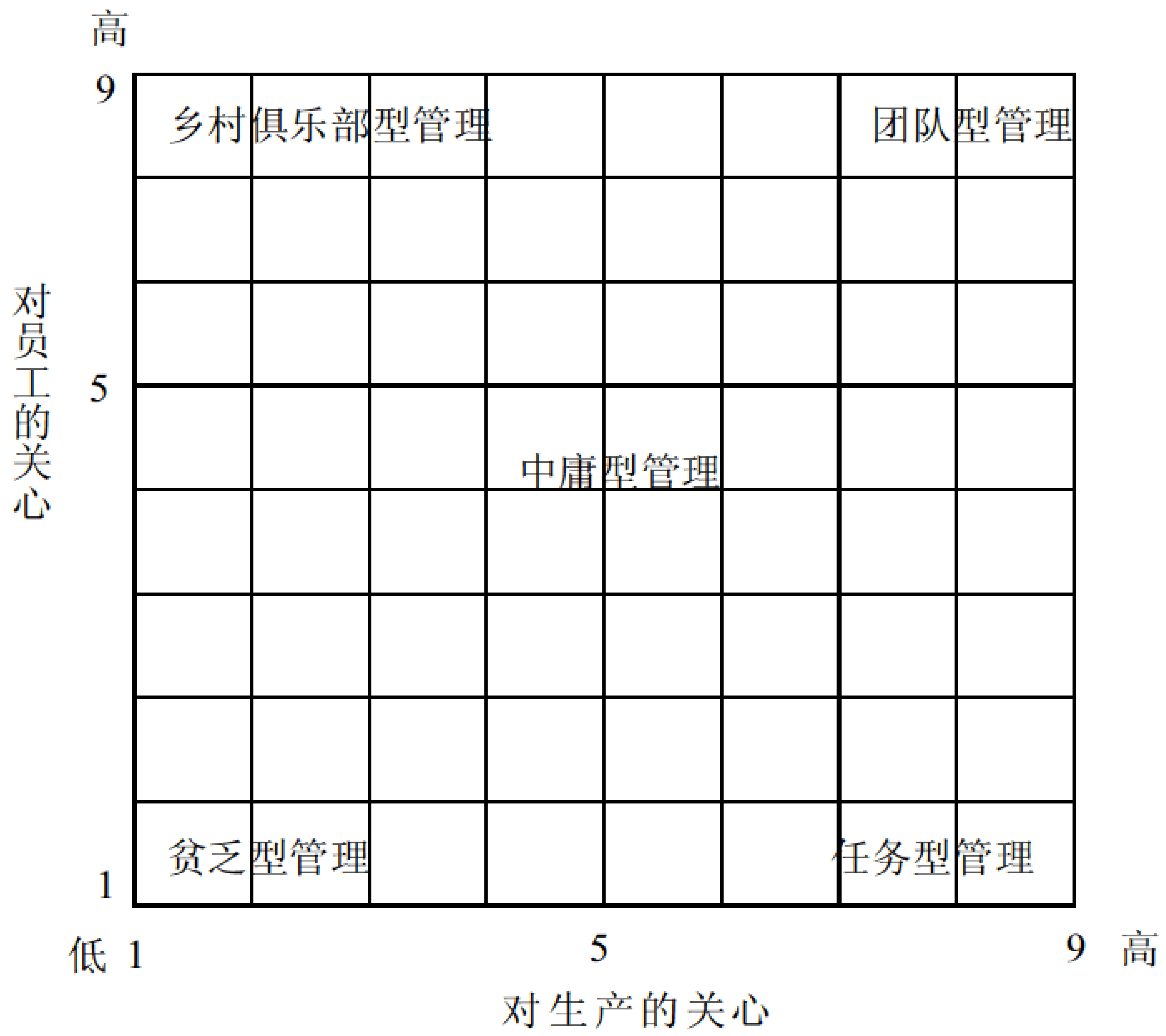


图 13.2　管理方格图

在图 13.2 的基础上，布莱克和莫顿概括出了 5 种典型的管理方式，并具体描述了它们的含义及特征：

9 • 1 型管理者：又称任务型管理者。这种类型的管理者非常关心生产，但不关心人。为了克服和预防组织人员对生产带来的不利影响，常常使用控制、统治、支配等权力形式。这类管理者往往固执己见，忽视或者压制不同意见。在短期内他们可能带来较高的生产效率，但由于不关心员工，不注意提高员工士气和保护员工的积极性，从长远看将会导致工作效率的下降。

1 • 9 型管理者：又称乡村俱乐部型管理者。这种类型的管理者只强调关心员工而不关心生产。高度重视友好关系的重要性，认为只要团队的气氛和谐、融洽就能带来好的绩效，只要管理者自己能得到员工的支持和拥戴，他（她）就是安全的。这类管理者倾向预见他人的欲望和要求，很少引起冲突。在这样的管理方式下，生产效率一般是不高的。

1 • 1 型管理者：又称贫乏型管理者。这类管理者既不关心生产，又不关心员工。具有这种管理方式的领导者经常逃避责任，回避义务，缺乏管理哲学和支配工作情境的志向和愿望，也无心去博得员工的喜爱与好评，得过且过，生产效率自然不会很高。

5 • 5 型管理者：又称中庸型管理者。这类管理者既要完成必要的管理任务，又要保持必要的士气，推崇对问题的折衷处理，追寻一种平衡式的解决方案，在追求的目标上，不去寻求对生产和员工都有利的最佳方案，而是寻求两者可以妥协的解决办法，把生产目标定位在员工和组织尚可接受的中间位置。

9·9型管理者：又称团队型管理者。这种类型的管理者既十分关心生产，又十分关心员工，把员工的愉快和舒适与达到工作目标放在同等重要的程度来关注。这类领导者不断鼓励、激励自己和他人做到最好，领导者的每个举措都是为了做得更好，而不只是为了使人感到愉快。当领导者对人和结果都高度关注的时候，如果有人滥用权力或者利用其他成员，领导者将会直面问题并能果断处理，从而避免给整个团队带来不利的影响。当生产与员工利益发生冲突时，领导者总是努力去寻找“双赢”的解决方案而不是折衷处理。

很显然，9·9型管理者是最佳的管理类型。研究也表明，具有这种管理风格的领导者的确比其他类型的领导者取得了更好的领导效果，既促进了生产效率，也更受员工的拥戴。

（三）PM领导行为理论

日本大阪大学的心理学教授三隅二不二在前人有关领导行为理论基础上，通过大量的调查与测量，于1958年提出了PM领导理论。

领导行为的PM理论是指，可以将领导方式分为两类：一类是以绩效为导向（performance directed）的领导方式，简称为P型领导；另一类是以维持群体关系为导向（maintenance directed）的领导方式，简称为M型领导。二者合起来，称为PM理论。

P型领导的行为特征是，将组织中每一个成员的注意力引向目标，使问题明确化，拟定工作程序，运用专业的评价标准来评定工作成果。

M型领导的行为特征是，维持和睦的人际关系，调解成员之间的纠纷，为群体成员提供发言的机会，促进成员的自觉性与自主性，增进成员之间的相互了解与交流。

为了测量被试的PM特征，三隅二不二设计了PM职能问卷。从工作激励、对待遇的满意态度、企业保健、心理卫生、集体工作精神、会议成效、沟通和绩效范围等八个方面出发编制问卷，进行5级评分，将被试所得的分数标注在一个两维的直角坐标系中，以此来确定此人的PM类型。为了达到测量的准确性，可以采用自评、互评、下级评价和上级评价的立体调查方法，以校正自我评价的偏差。

根据PM两种行为特征的得分分布，可以把领导行为分为四种类型，即PM型、P型、M型和pm型，如图13.3所示。

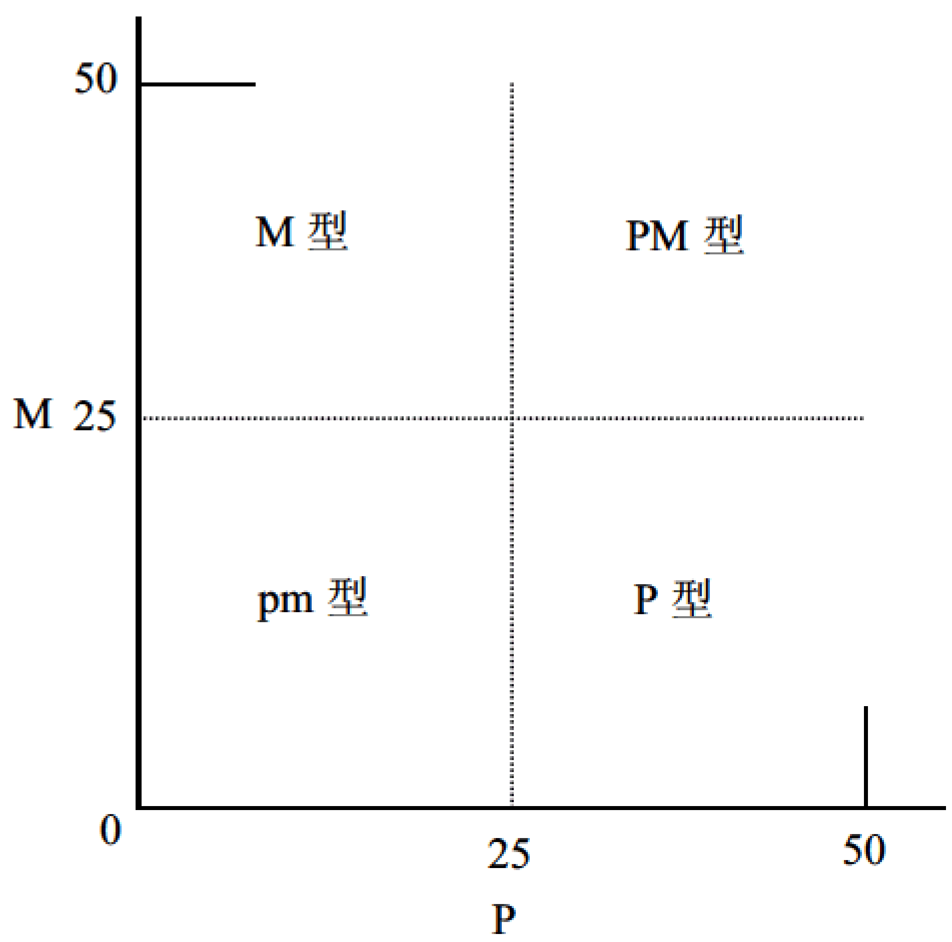


图13.3　PM领导类型图

三隅二不二教授进一步研究发现，这四种类型领导者的管理效果是不同的。PM 型领导者的效果最好，可导致最高的生产效率，员工对领导者的高度信赖，及领导与下属之间的亲和力。P 型领导者和 M 型领导者只能取得中等的生产效率。pm 型的领导者效率最差。

我国学者凌文辁等人（1985）在三隅二不二研究基础上，结合我国的文化特点，根据大规模的问卷施测结果，采用因素分析和聚类分析等方法，提出了领导行为的三因素结构模型，即中国人领导行为的CPM模型。该模型由三个维度构成：个人品德因素C（Character）、工作绩效因素 P（Performance）、团体维系因素 M（Maintenance）。

三、领导有效性的权变理论

20 世纪 60 年代以后，关于领导有效性研究转入权变理论（或情境理论）阶段。权变理论认为，领导的有效性不是取决于领导者不变的品质和行为，而是取决于领导者、被领导者和情境条件三者的配合关系，即领导有效性是领导者、被领导者和领导情境三个变量的函数。

领导有效性＝f（领导者，被领导者，领导情境）

权变领导理论把领导看作一个动态的过程。由于特质理论不能准确地预测领导者的行为，甚至难以解释不同情境下领导者行为的多样性，而行为理论在解释某些领导行为时又显得过于简单，有时甚至难以自圆其说，如对同一种领导行为为什么在不同的群体中会产生不同的效果等问题不能给出圆满的解释，因此研究者把注意力转移到了领导情境方面。影响领导有效性的大量情境因素逐渐被识别出来，于是研究者试图将这些情境变量整合起来，形成了多种权变领导理论模型，比较有代表性的有：领导有效性权变模型、路径—目标领导理论、生命周期理论。

（一）菲德勒的领导有效性权变模型

菲德勒（F.Fiedler）经过长期的研究，提出了领导有效性的权变模型。他认为，任何形态的领导方式都可能有效，关键在于领导风格和具体组织情境的匹配程度。领导效果的好坏取决于三个维度上的条件：领导者与被领导者关系、任务结构以及职位权力。如果这三个维度上的条件都好的话，情境对领导是有利的。具体地说，如果领导者被追随者接受和尊敬（第一个维度），任务高度结构化，即每件事情都可以描述和有程式化的运作方式（第二个维度），且领导者的职位权力和权威非常正式化且稳固（第三个维度），那么这个情境对领导者是有利的。但是，如果出现相反的情况（三个维度都低），那么情境对于领导者而言就是非常不利的。菲德勒通过他的研究证明，情境有利和领导风格共同决定了领导有效性。总起来看，在非常有利和非常不利的情境下，任务导向或者顽固、独裁型的领导者是最有效的；当情境只是适度有利时，人性取向或民主型的领导者是最有效的。情境变量与领导形态之间更加复杂的关系如表 13.2 所示。

表 13.2　菲德勒领导形态与情境变量之间的关系

情境	领导者与被领导者关系	任务结构	职位能力	有效领导形态
1	良好	有结构	强	任务导向
2	良好	有结构	弱	任务导向
3	良好	无结构	强	任务导向
4	良好	无结构	弱	人群关系导向
5	不良	有结构	强	人群关系导向
6	不良	有结构	弱	人群关系导向
7	不良	无结构	强	任务导向
8	不良	无结构	弱	任务导向

为了确定一位领导者的领导风格和倾向，菲德勒开发了一种名为“最难共事者量表（least-preferred-coworker scale, LPC）”的工具，通过测定领导者的 LPC 分值，可以识别出领导者的激励层次。低 LPC 领导者主要受到任务的激励，这意味着这些领导者主要从完成任务中获得满足感，但如果任务以可以接受的方式完成了，则低 LPC 领导者会移向第二个激励层次，即建立和保持与追随者的关系。高 LPC 领导者主要受到关系的激励，这意味着这些领导者主要从建立和保持密切的人际关系来获得满足。如果高 LPC 领导者已经与追随者建立了良好的关系，则他们会移向第二个层次，即完成任务。把领导者的 LPC 得分、情境有利性和领导方式选择结合起来，菲德勒建立了如图 13.4 所示的领导者 LPC、领导风格与情境交互作用的领导有效性权变模型。

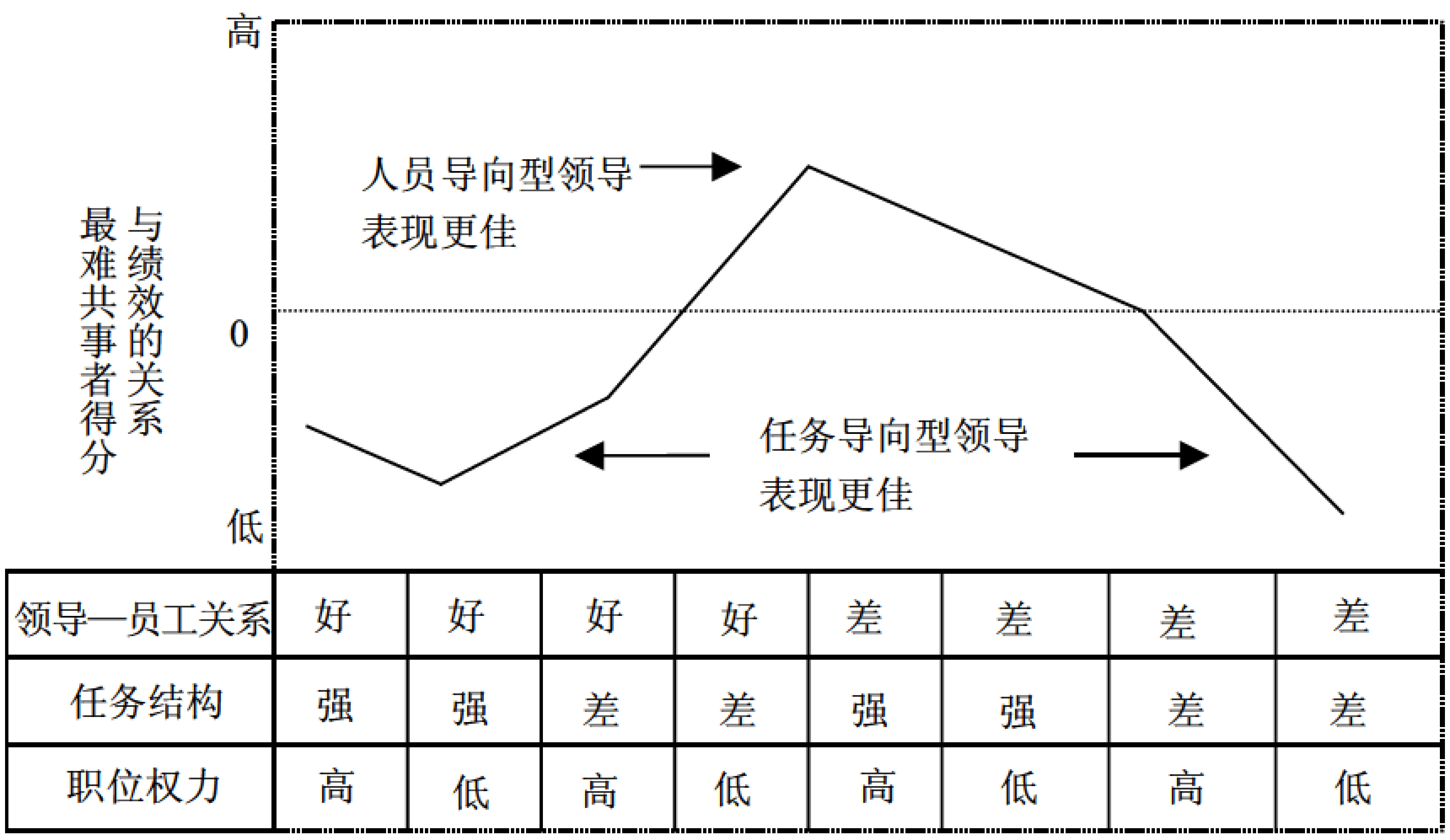


图 13.4　菲德勒的领导权变模型

那么，为什么任务导向型领导者在非常有利情况下是成功的？菲德勒的解释是：在非常有利的情境中，领导者拥有权力、非正式的后援、相对结构化的任务和准备好接受领导

的组织，且组织成员期望被告知应该做什么。在这种情况下，不必与全体成员讨论，任务导向的领导者只需要发号施令，组织就能按照既定方向发展。此种情境下，领导者明智的选择是：宁愿做出一个错误决定，也比什么都不做好。勇于负起责任，并做出完成任务所必须的决定的领导者，常常是有效的领导者。

在相对不利的情境中，领导者不被团体成员完全接受，任务也不是完全结构化的，并且领导者仅仅被授予了部分权力和权威。菲德勒预测，此种情境下采用人性取向、民主取向及关系取向的领导者是最有效的，这一结论已经被相关研究所证实。

菲德勒的领导权变模型既吸收了以往理论关于领导风格或类型的观点，又引入了情境变量，试图在二者之间建立起耦合关系，这对于解释领导行为的多样性及其有效性具有更加适贴的意义。该模型的可操作性较强，对于何种情境适合何种类型的领导方式给出了详细的说明，对领导者的选拔和任用具有切实的指导意义。正因为如此，它引发了随后一系列的研究。

（二）豪斯的路径－目标领导理论

路径—目标领导理论是由加拿大多伦多大学豪斯（Robert House）教授首先提出来的。它是从动机期望理论框架中分离出来的另一种被广泛认可的权变理论。路径—目标理论试图解释领导者行为对下属动机、满意度和绩效的影响，也就是说，它试图弄清楚领导者如何做才能为追随者铺平通向目标的路径，以使追随者达到所期望的结果。为了回答这个问题，豪斯首先总结出了四种主要的领导方式，即指导型领导、支持型领导、参与型领导和成就取向型领导。

指导型领导：下属准确地知道领导者期望他们做什么，而且在做的过程中领导者还会给予具体指导，不需要下属参与做什么和怎么做的决策。

支持型领导：领导者是友好的并且是可亲近的，仅对下属的行为表现出诚恳的关注。

参与型领导：领导者寻求并且采纳下属们的建议，但仍然由领导者做出决策。

成就取向型领导：领导者为下属设定有挑战性的目标，并给予充分的鼓励，让他们自己设法达到这些目标。

菲德勒认为，领导者的风格或类型一旦形成之后就不易改变，因此只能根据不同的组织情景选用不同类型的领导者。与此不同的是，豪斯的路径—目标领导理论认为，上述各种不同的领导方式并非固着于单一领导者身上，而是可供同一个领导者在不同情境中选用的行为模式。也就是说，同一个领导者能够并且确实在不同的情境中使用不同的领导方式。豪斯关心的问题是在何种条件下上述何种领导方式可以为下属铺平通向目标的道路。在这个问题上，豪斯再一次表现出了与菲德勒的不同。他除了承认菲德勒所强调的环境变量之外，又加入了追随者的个人特征变量，即追随者的个人特征和组织环境特征共同决定了下属对各种领导方式的知觉，而正是下属对领导行为的知觉影响了他们的满意度、角色和目标清晰度及随后的绩效表现。豪斯所构建的从领导行为到目标实现的权变路径如图 13.5 所示。

依据图 13.5 所示的模型，领导者需要视追随者特征和具体组织环境的变化而不断改变或调整自己的领导方式。无论采用哪种领导方式，领导者的目的都是为了影响追随者的知觉和动机，引起良好的心理状态和行为反应，为实现个人和组织目标扫平道路。

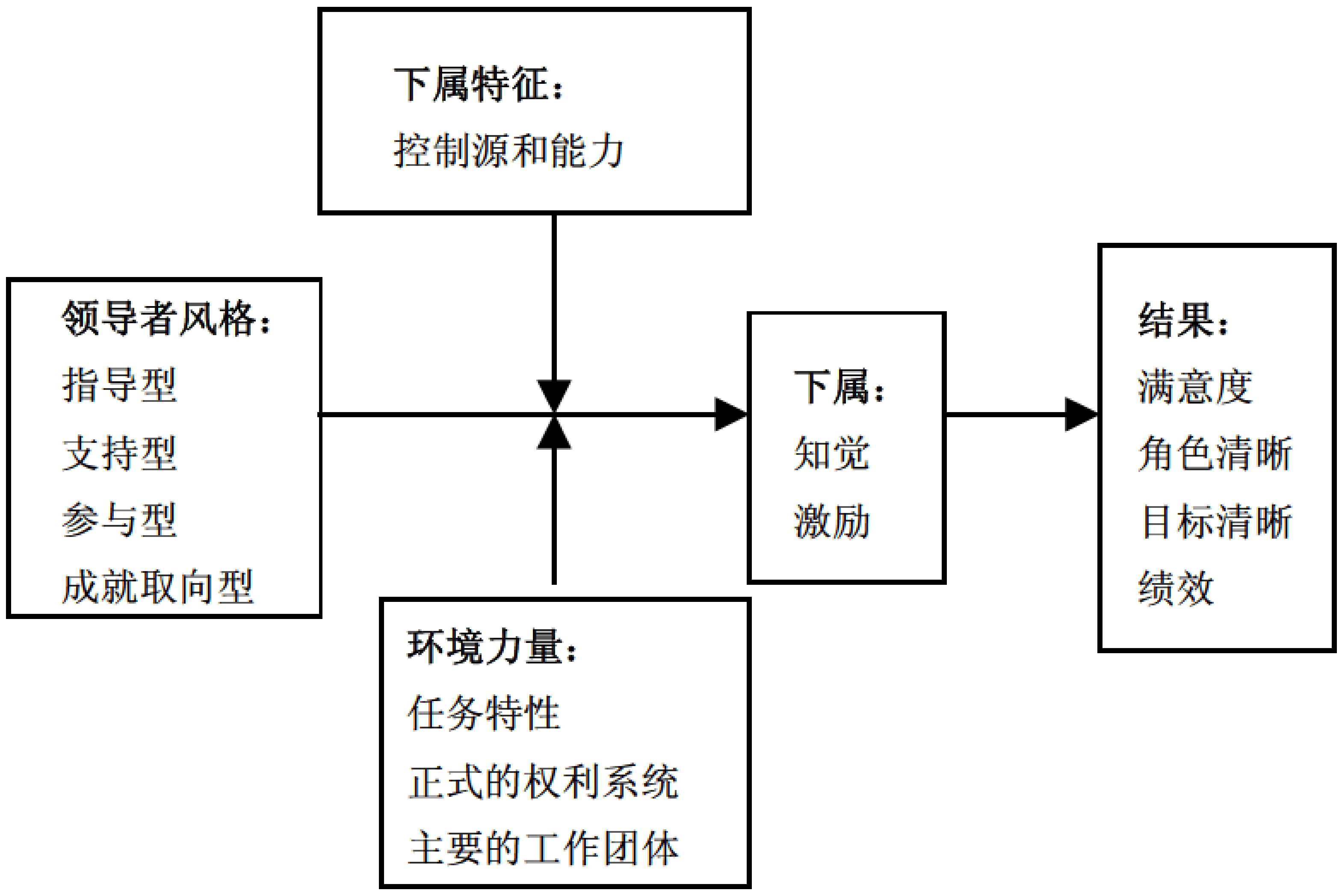


图 13.5　路径—目标关系模型

（三）卡门的生命周期理论

该理论由俄亥俄州立大学的卡门（Kaeman）教授提出。他的主要观点是：领导者的行为要与被领导者的成熟度相适应，即随着被领导者的成熟度逐渐提高，领导者的领导方式也应做相应的改变。所谓被领导者的成熟度主要是指心理的成熟度，包括成就感、有工作经验和受过良好的教育等。年龄是成熟度的一个因素，但并非最重要的因素。员工的成熟度有一个发展的过程，即由不成熟、初步成熟、比较成熟最终发展到成熟。因此，领导者的领导方式也要随之发生改变。卡门区分出了授权式、参与式、说服式和命令式四种主要的领导方式，并依照被领导者的成熟度不同将它们置于如图 13.6 所示的关系和任务导向双维坐标图中，用来说明不同类型的领导者在员工处于何种成熟度时采取何种领导方式更为有效。

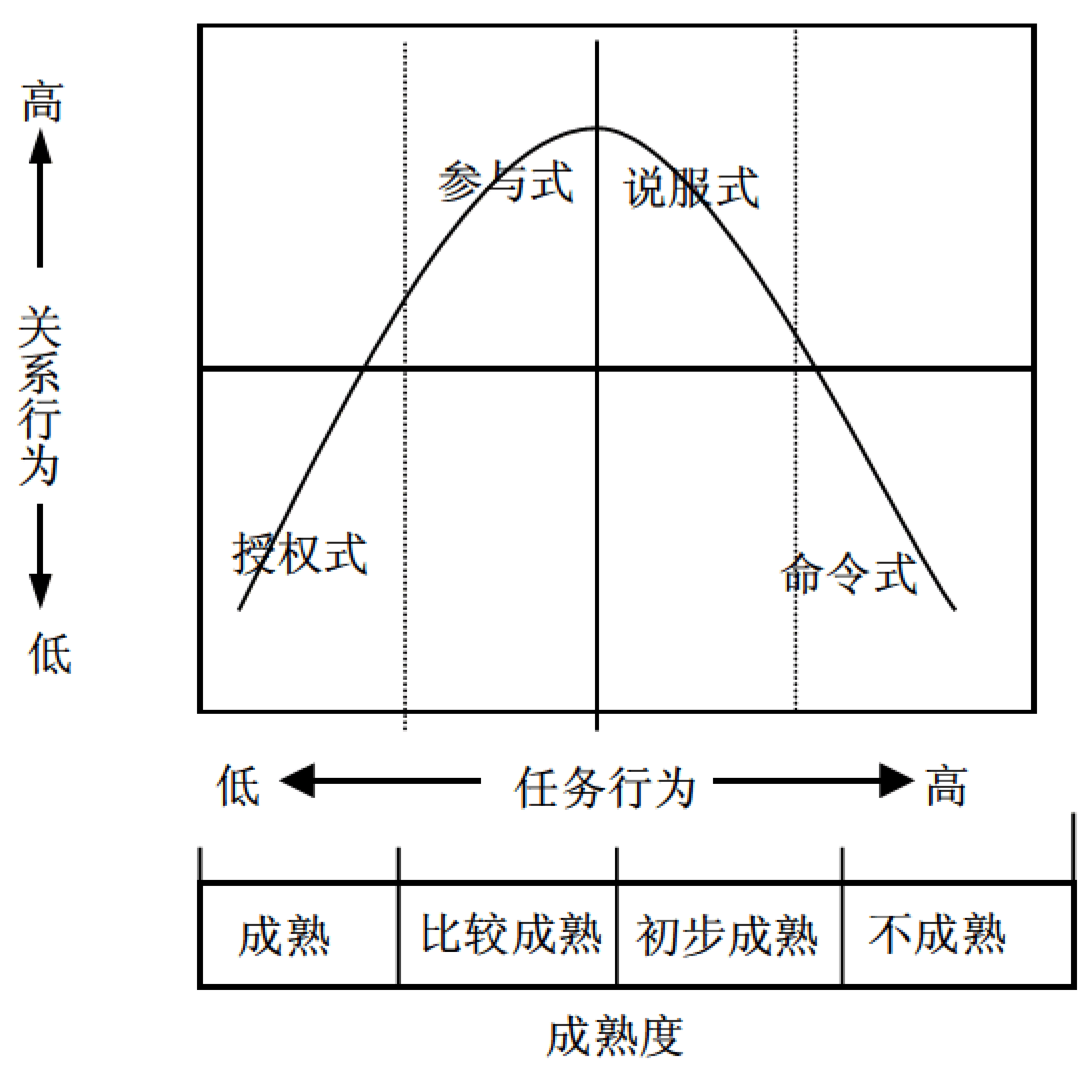


图 13.6　生命周期理论模型

值得注意的是，生命周期理论认为，“高工作、高关系”的领导者不一定总是有效的，同样，“低工作、低关系”的领导者也不一定总是无效的，关键是要看所采取的领导方式是否与被领导者的成熟度相匹配。

第三节 当代领导有效性理论

进入 20 世纪 80 年代以后，西方组织行为学界对领导理论的研究进入了大发展的时期，众多的领导理论研究学派纷纷登场，研究领域不断拓宽和深入，研究成果及论文呈爆发之势，但目前还没有形成整合的态势。此时期比较有代表性的领导理论主要有以管理学家巴斯（Bernard M. Bass）为代表的魅力领导理论、以菲德勒和乔·葛西亚为代表的领导认知资源理论、以伯恩斯和巴斯（Mac G.Burns & Bernard M. Bass）等人为代表的交换型领导和变革型领导理论、以克尔和杰米尔（Kerr & Jermier）为代表的领导替代理论以及以格拉斯和斯腾伯格（Graves & Sternberg）等人为代表的内隐领导理论等。

一、魅力型领导理论

魅力型领导理论的研究起源于对行政领导行为、社会活动和宗教仪式上的领导行为的关注。最近几年，组织中领导者的魅力越来越引起人们的兴趣，它被称为“可以点燃追随者的精力和许诺的一把火”。所谓魅力，是指能对追随者们产生深刻而非凡影响倾向的个人能力或力量。领导魅力能够产生超越职责和组织的力量和结果。有魅力的领导者具有鼓舞和激发员工斗志的能力，能够对人们特别是追随者产生一种情感上的影响，感染人们的心灵和思想，甚至能够使追随者产出高于期望的绩效。

现代魅力型领导理论的研究和发展应归功于加拿大管理心理学家罗伯特·豪斯（Robert House）。他基于对政治和宗教领袖的分析，提出了魅力型领导者应具有的特征：（1）具有自信且信任下属；（2）对下属有高度的期望；（3）有理想化的愿景；（4）极具个性化的人格特征。豪斯指出，魅力型领导的追随者们认同领导者和他们下达的任务，显示出对领导者的无限忠诚和信心，效法领导者的价值观和行为，并从他们与领导者的关系中获得自尊。

美国管理学家巴斯（Bernard M. Bass）以杰克·韦尔奇、比尔·盖茨及其他一些杰出的女性领导者为研究对象，扩展了魅力型领导研究的范畴。他注意到，魅力型领导者一般具有出众的辩论和说服技巧，有技术专长，并且能够促进追随者们在态度、行为和情感上的变化。

美国《财富》杂志上的一篇文章以幽默的方式生动地描述了一位有魅力的领导者所具有的特点：“他上了一个二流大学，没有获得过 MBA 头衔。但是他有一种气质。他说服人们去做他们不想做的事情。人们为他铲平山丘，冲过火海，赤脚在玻璃上行走……”

美国南加州大学马歇尔商学院的杰伊·康格（Jay A. Conger）等人认为，具有超凡魅力的领导人和不具有超凡魅力的领导人存在性格差异，主要表现在表 13.3 所示的几个方面。

表 13.3　具有超凡魅力的领导人和不具有超凡魅力的领导人的性格差异

性 格 维 度	不具有超凡魅力的领导者	具有超凡魅力的领导者
亲切度	分享观点从而使领导亲切	分享观点和理想愿景使得领导亲切，并成为值得尊敬的英雄
可信赖程度	对说服中的说话艺术不感兴趣	会花费大量的成本和面对极大风险进行热情的说服
和现状的关系	试图保持现状	创造革新的气氛
未来目标	将目标限制在与现状差别不大的范围	与现状有很大差别的理想
清晰度	对领导的目标和动机不是很清晰	对领导的前景和动机有很清楚的认识，并且具有灵感
竞争	在已经存在的结构中采用可获得的方法实现目标	采用超出现存秩序的非传统的方法
行为	传统的，与公认标准一致的	非传统的，与公认标准相反的
影响	基于来自于职务的权力	超越职位，来自于专家权力以及对领导的尊敬和赞赏的个人权利

管理学家理查德 L.达夫特（Richard L. Daft）在《领导学》一书中总结了具有超凡魅力的领导人应具有的三个典型特征：

（1）具有超凡魅力的领导人创造一个革新的氛围，对未来的理想有清醒的认识，而这个理想是远远优于现实的。他们有能力用一种清晰的强有力的方式和别人交流复杂的主题和目标，从而使上至副总经理下至看门人都能理解和明确自己的任务。具有超凡魅力的领导人向追随者灌输一种持久的信念，即使信念不能在当事人所实现的很容易的目标中得到体现，信念本身就成为对追随者的奖赏。

（2）具有超凡魅力的领导人愿意冒巨大的个人风险来获得大众的信任。他们用热情的谈话技巧把自己定位为一个具有超凡魅力的领导人。通过面对风险，领导者能够增加自己对追随者的情感上的吸引力。

（3）具有超凡魅力领导人的影响力来自于个人力量而不是职位力量。人们喜欢领导者并且想成为像他那样的人。追随者尊敬和钦佩领导者，是因为他们的知识、经验或者是个人魅力，而并不是他们在组织中的名衔和地位。尽管具有超凡魅力的领导人可以拥有职位权力，但他们超越了这种正式的职位权力，因为他们的影响力是基于个人品质而不是企业赋予他们的权力。

二、交换型与变革型领导理论

交换型领导理论认为，领导过程就是领导者和追随者的一种交易过程。交换型领导者

确认追随者的需要和要求，并告知他们如何通过实现某种目标或者履行某种职责来达到和满足这些需要和要求。追随者通过工作绩效而得到奖励，领导者则从任务的完成中获益。交换型领导者关注的是现在，擅长让企业平稳有效率地运作，领导者擅长计划、预算等传统管理职能，他们通过阐明期望，帮助追随者建立自信，满足追随者的需要，提高员工士气。他要保持组织的稳定性而不是促使组织变革。

变革型领导者的特点是有能力带来巨大的变化。他们引导企业的愿景，制定企业的战略，营造企业的文化。他们关注的是企业愿景、共享价值等无形的品质，而不是用规定的、指导和控制的渠道与追随者沟通。变革型领导是基于领导者的个人价值、信念和品质而不是领导者和追随者的交换过程，其特点主要有四个方面。

（1）变革型领导者试图将追随者发展成为领导者。追随者被赋予更大的自由度来控制他们自己的行为。变革型领导者将人们集结在任务周围，鼓励追随者对问题进行争论，帮助人们用新的方式看待事物，由此引发组织变革。

（2）变革型领导者把追随者的需求由低层次的物质需求提升到高层次的精神需求。低层次的需求可以通过工资、安全的工作环境等条件来满足。变革型领导更注重追随者的成长和发展需要。因此，只有把追随者的需求和能力提升到一个更高的档次，并把他们和企业的任务联系起来，通过改变追随者来改变企业，才能达到组织变革的目的。

（3）变革型领导者激励追随者牺牲自己利益而为团队的利益而努力。他们让追随者清楚地认识到目标和产出的重要性，鼓励追随者为了组织的长远目标而放弃个人的短期利益，强调信任的重要性。变革型领导者不仅用他们的人格魅力来激励追随者，也要使追随者相信变革的重要性，并且能做到为了伟大的目标而心甘情愿地做出牺牲。

（4）变革型领导者善于描绘一幅理想未来的愿景，并把它传递出去，让人们相信变化的阵痛是值得的。他们重要的角色是找出并显示优于现在的愿景，获得人们的支持，共同为梦想而努力。只有当人们清楚地知道企业要实现的愿景时，变革才是有可能的。

交换型领导保持稳定性，而变革型领导追求为企业创造未来。实证研究表明，有效的领导者显示出交换型和变革型的双重形态，他们不仅重视建立愿景和赋予追随者动力，而且重视设计用来实现愿景的结构、控制系统和奖励系统以及有效的沟通方式。调查表明，许多有成就的领导者都体现了变革型和交换型的平衡和统一。

三、社会认知资源理论

1987 年，菲德勒和乔·葛西亚试图解释领导者通过什么获得有效的团体绩效。为此，他们考察了认知资源，诸如智能和经验等与组织绩效相关的条件，提出了社会认知资源理论。这个理论基于两个基本假设：其一，睿智的、有才干的领导者比德才平庸的领导者能制定更有效的计划、决策和活动策略；其二，这类领导者有能力通过指导行为传达他们的计划、决策和策略。据此可以做出三项预测：（1）在支持性、无压力的领导环境下，指导

性行为只有与高智力结合起来，才会导致高绩效水平；（2）在高压力环境下，工作经验与工作绩效之间呈正相关；（3）在领导者感到无压力的情境中，领导者的智力水平与群体绩效呈正相关。

按照认知资源理论，团队的绩效水平取决于两种领导品质（智能和经验）、两种领导行为（指导型领导和参与型领导）以及两个方面的领导情境（人际关系压力和组织团体的性质）之间复杂的相互作用。其主要因果关系如图 13.7 所示。

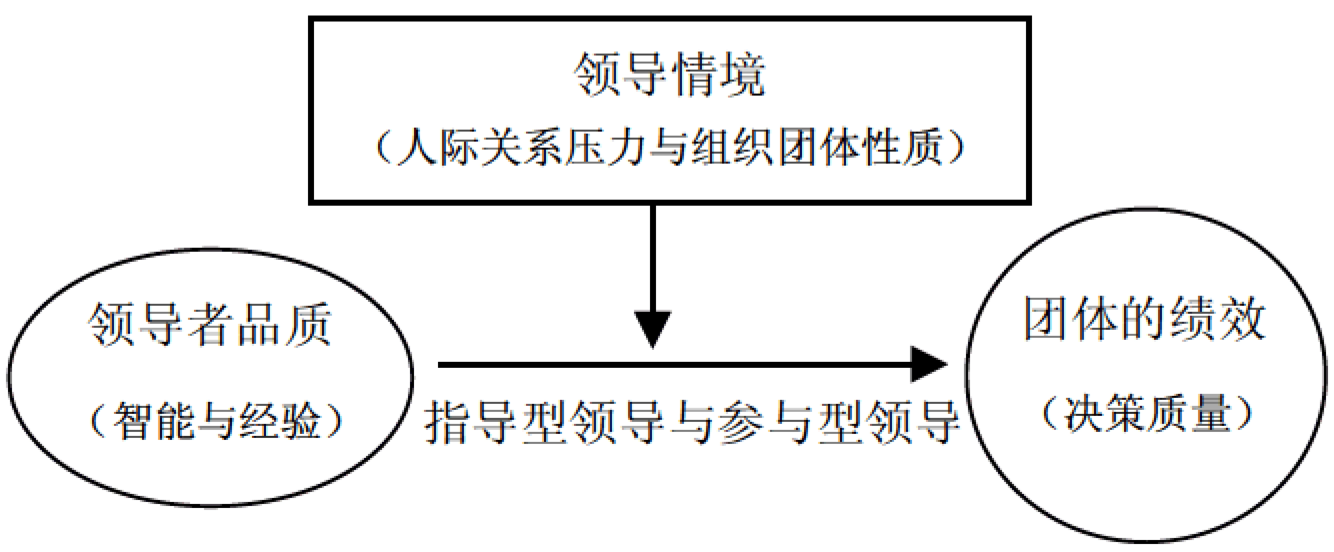


图 13.7　认知资源理论的主要因果关系

菲德勒的研究表明：

（1）领导者的人际关系压力调节着领导智能与部属绩效之间的关系。压力可能来自于组织层面，如角色冲突、缺乏资源和支持等，也可能来自于团体内部，如工作危机、领导者与部属的冲突等。低压力情形下，高智能导致好的计划和决策；高压力情形下，领导者智能与决策质量之间不存在相关。

（2）领导者的人际关系压力调节着领导经验与部属绩效之间的关系。人们在高压力之下倾向于重复以前的行为去处理问题，而不是将它作为新问题处理。对于有经验的领导者来说，在低压力情景下主要依赖智力去决策，在高度的人际关系压力情境下，经验与领导者的决策质量呈正相关。对于没有经验的领导者来说，两种情形都依赖于智力进行决策。

（3）只有当领导是指导型的、并且部属需要指导去完成任务时，领导者的智能和经验才共同对组织绩效有贡献。

四、领导替代理论

权变领导理论侧重于对领导的类型、追随者的特性以及环境的特征进行综合研究。最新的研究指出，环境因素的变化往往是极其重要的，它常常取代了领导，或者削弱了领导的作用。克尔和杰米尔（Kerr & Jermier）通过实证研究得出了这样的结论：环境变量总是趋向于取代领导或者削弱领导的作用。

克尔和杰米尔区分了替代领导的三类变量，即组织特性、任务特性和追随者特性。它们分别可以替代或者削弱领导者的不同作用，如表 13.4 所示。

表 13.4　三类变量对领导作用的替代与削弱

变　　量	任务导向型领导	关系导向型领导
组织因素：		
有凝聚力的组织	替代	替代
低职位权力	削弱	削弱
正式化（角色、程序）	替代	无
死板（规则、政策）	削弱	无
工作地点分散	削弱	削弱
任务特征：		
任务高度结构化	替代	无
自动反馈	替代	无
内在满意任务	无	替代
追随者特性：		
职业、专业导向	替代	替代
经验、能力、训练	替代	无
漠视奖励	无	削弱

（资料来源：The Leadership Experience，2nd Edition，2002）

克尔和杰米尔的研究表明，在某些特定的情境下，领导的作用是可以代替或者削弱的。从组织和团队层面上看，在有详细的规则、制度和政策的组织中，当部属掌握了行为规则和政策时，就很少需要指导。如果规则和政策是非常刚性的，就阻止了试图改变工作分工和工作程序的努力。当部属的工作地点分散时，如销售代表的工作，任务型的指导也是无力的。另外，对于凝聚力强的组织而言，领导更加显得可有可无，因为团结的工作群体不需要领导努力去鼓动下属和调配任务。从任务特性看，日常性的任务、结构化的任务、简单的任务是可以替代任务型领导的，如果任务本身能够提供是否完成的自动反馈，领导就不必提供更多的反馈。如运用计算机网络系统工作的员工就不需要更多的监督。另外，对于部属感兴趣的工作也无需领导者进行支持、引导与鼓励。从部属特性看，当部属有丰富的经验和足够的训练时，他们已经拥有做什么和如何做的知识和技能，需要更少的指导，如药剂师、飞机领航员、会计师等。同样，专业工作者受到他们的价值观、需要和道德的内在激励，也不需要领导者的鼓励就能提高工作质量。此外，如果部属希望花更多的时间与他们的家人在一起，他们就会对超时间工作而获得的金钱或其他奖励不感兴趣。

五、内隐领导理论

前面介绍的领导理论主要关注领导者的特质、行为及领导者所处环境的特征对领导有效性的影响。尽管也有一些理论涉及到了被领导者的特征，但并未对此进行深入分析和探讨。实际上，如前所述，领导本身就是领导者和被领导者之间在特定组织情境中交互作用

和影响的过程。被领导者如何看待领导者、如何理解领导者的行为、对领导者作出什么样的反应，直接影响领导的有效性。正是出于这种考虑，格拉斯和斯滕伯格等人于上世纪 70 年代提出了内隐领导理论的概念，揭开了内隐领导理论研究的序幕。此后，该领域的研究此起彼伏，逐渐集中在以下几个问题上。

（1）内隐领导理论的概念：内隐领导理论是指人们特别是组织中的被领导者“内心”关于领导者和领导行为的概念、看法、观点和信念体系。按照格拉斯和斯腾伯格的说法，内隐领导理论是人们“内心”关于领导概念的结构，它以某种形式已经存在于人们的头脑之中，既涉及领导者是什么的问题，又涉及领导者应该是什么的问题。人们对领导行为的感知、描述和评价将受到这种内隐领导理论的影响。简单地说，内隐领导理论就是被领导者持有的、隐藏在其内心深处的领导理论。

（2）内隐领导理论的因素构成：关于内隐领导理论的因素构成，国内外学者进行了大量的探讨。在国外，格拉斯（1982）发现了美国人内隐领导理论的三维结构，即工作热情、参与和激励。奥夫曼（Offerman，1986）发现了美国人内隐领导理论的八维结构，即感受性、献身精神、专制、魅力、吸引力、男性气质、智力和力量。霍姆伯格和阿卡伯姆（Holmberg & S.Akerblom,2000）发现了瑞典人内隐领导理论的 12 个结构主题，即表现、行动、有魅力、有预见性、诚实、谦逊、实用主义、团队建设、平等、共识、企业家观念及程序化，并提出了杰出领导者的六大主题，即行动、合作、平等、表达能力、热心和激励。

关于中国人的内隐领导理论研究起步较晚，研究成果也较少。从目前掌握的资料来看，有代表性的研究成果主要有我国台湾学者郑伯壎、刘兆明对华人内隐领导行为与工作动机关系的研究，以及我国内地学者凌文辁等人建构中国领导行为 CPM 模型的研究。郑伯壎（1981）将美国俄亥俄州立大学的“督导行为描述问卷（SBDQ）”翻译成中文，对华人企业员工进行施测，因素分析的结果发现华人对领导行为的认知中除了原有的“体恤”及“结构”两因素外，可能尚存有一个“公正严明”的因素。刘兆明（2000）通过自编的领导行为问卷对华人组织中的员工进行施测，因素分析的结果得到了“正性行为”和“负性行为”两大因素，他因此认为华人的内隐领导认知结构是包含“正性行为”和“负性行为”的两极性结构。

内地学者凌文辁等人（1985）根据大规模的问卷施测结果，采用因素分析和聚类分析等方法，提出了领导行为的三因素结构模型，即中国人的领导行为 CPM 模型。该模型由三个维度构成：个人品德因素 C（character）、工作绩效因素 P（performance）、团体维系因素 M（maintenance）。在此基础上，凌文辁等人（1988）编制了“中国领导特质因素问卷”，对企业员工进行测试，调查数据和因素分析的结果发现，中国人内隐领导理论由个人品德因素、目标有效性因素、人际能力因素和多面性因素四个维度构成，为此，凌文辁建构了中国人内隐领导理论的四维度结构模型。凌文辁的博士研究生林琼（2003）使用“中国领导特质因素问卷”对深圳、广州、杭州等地的企业员工进行再调查，因素分析仍然发现以上四个维度的内隐领导认知结构，只是各因素的负荷量与 1988 年的结果之间存在差异。

（3）内隐领导理论的来源：关于内隐领导理论的来源及形成机制，目前的研究还较少。洪特等人（Hunt，Boal & Sorenson ）注意到，一些背景经验会影响内隐领导理论的形成，

如父母特征和童年经历。郝尔等人（Hall & Lord）认为，稳定性、延续性、倾向性等人格特质可能直接地关系到个体对领导的认知。另外，工作场所领导与下属的相似性及相互作用的方式也影响内隐领导理论的形成。

（4）内隐领导理论的作用：内隐领导理论无论对个体还是组织而言都是有益的。劳德和马塞尔（Lord & Maher）发现，人们应用内隐领导理论，既作为解释他人（特别是领导者）日常行为的基础，也作为产生自己行为的基础。从转换型领导的观点出发，巴斯发现，典型的内隐领导理论特质与转换型领导的内省得分高相关。

内隐领导理论更重要的作用还在于其影响工作场所中领导者与追随者的相互关系，未来的研究需要探测内隐领导理论如何影响没有经验的新员工，使其形成或者恢复组织知觉。例如，强刺激环境（如军事）可能培养共同的理想化领导概念，而弱刺激情境中个体差异将更加显现。此外，自我确认理论认为，内隐领导理论涉及到自我概念，个体试图从别人那里得到领导形象的信息，来确认他们的自我概念。有理由假定，组织通过塑造强组织文化模式可以引导员工的自我概念，从而改善追随者的内隐领导理论。

内隐领导理论研究的许多发现对管理实践有着重要的意义。首先，下属倾向于观察领导者身上表现出来的与自己头脑中的领导者原型接近的人格特征。领导者对拥有不同内隐领导理论的下属提出同样的要求将会起到不同的作用。第二，在团队成员的选择方面，管理者可能更喜欢那些持有与自己相似内隐领导理论的成员，而排斥具有不同领导概念的个体。这暗示着对多样性的组织而言，仅仅与领导相似的成员可能被提升。第三，哪些领导者特征或行为能够激发追随者最大的承诺？答案是：与追随者头脑中的内隐领导理论相似或接近的特征和行为直接影响下属的信任感和满意度，从而容易激发追随者的承诺。

本章小结：

1. 领导是指引和影响个人或组织，在一定条件下，实现某种目标的行动过程。实施指引和影响的人称为领导者。领导的本质是人与人之间的一种互动行为。

2. 领导者在领导活动中所表现出来的行为就是领导行为，领导行为的影响和作用称为领导功能，通常领导的功能表现在组织功能、激励功能和角色功能三个方面。

3. 传统的领导有效性理论可分为领导的特质理论、行为理论和权变理论。特质理论以心理学家吉伯、爱德温•吉色利等的研究为代表，强调领导者自身要有一定数量的、独特的并且能与他人区别开来的品质与特征；行为理论包括领导四分图模式、管理方格理论以及 PM 领导理论，强调领导者的行为方式或类型的重要性；权变理论以领导有效性权变模型、路径—目标领导理论、生命周期理论为代表，强调领导者、被领导者和情境条件三者的配合关系。

4. 当代领导理论包括魅力型领导理论、交换型与变革型领导理论、社会认知资源理论、领导替代理论和内隐领导理论等。魅力型领导理论强调领导者身上能对追随者们产生深刻而非凡影响的个人能力或力量；交换型领导者按照与追随者之间的交易关系处理和解决问题，变革型领导者基于个人的价值、信念和品质处理和解决问题；社会认知资源理论强调

领导者的智力和经验对团队的绩效的影响，并重视领导者品质、领导行为和领导情境之间存在的复杂关系；领导替代理论重视组织特性、任务特性和追随者特性对领导作用的替代或者削弱；内隐领导理论强调被领导者“内心”存在的关于领导者和领导行为的概念、观点和信念对领导有效性的影响。

本章思考题：

1．简述领导的概念，分析领导和领导者的联系与区别。
2．简述领导特质理论的主要内容，分析其合理性和局限性。
3．简述领导行为理论的主要观点，举例说明其合理性与局限性。
4．简述菲德勒领导权变理论的要点，分析其合理性与局限性。
5．简述领导生命周期理论的主要观点，举例说明其合理性和局限性。
6．当代领导理论有哪些？简述各自的主要内容。
7．举例说明领导特质理论、行为理论以及权变理论在现代组织管理中的应用。

第十四章　领导策略与方法

第十三章介绍了领导有效性的理论，即关于什么东西决定领导效果的一些观点。实际上，无论有什么样的领导理论，也无论有多么好的领导者，如果不能将好的领导理念或领导特质及行为很好地落实到具体行动上的话，也很难达到理想的领导效果。好的领导理念或领导特质及行为是通过领导者的一言一行、一举一动实现出来的。领导是需要讲究策略和方法的。本章拟对领导过程几个重要环节上的策略和方法问题加以探讨。

第一节　领导者人力资源管理的策略与方法

一、领导者授权的策略与方法

（一）什么是授权

简单地说，授权就是领导者向下属下放权力和责任的一种管理方法。授权意味着作为下属的员工获得了领导者的授意，可以自主地完成特定任务，获得相应的利益，同时也承担相应的责任。授权也意味着领导者将决策权赋予那些最有可能受到决策影响并将实施该决策的人。授权更关注自主性、责任以及下属的发展，而不仅仅是参与。研究表明，善于授权的领导者往往有更高的业务绩效。威尔考克斯（Wilcox）认为，与不能授权的领导者相比，有授权技巧的领导者拥有满意度更高的下属和绩效更佳的工作团队。因此，掌握有效的授权技巧与艺术，是领导者必备的技能之一。

（二）授权的意义

对组织的领导者来说，有效授权的重要意义体现在以下几个方面：首先，授权将领导者从烦琐的杂务中解放出来，为领导者从事更重要的活动节省了宝贵时间。领导的本质就是通过与其他人的共同努力实现目标，而不是大包大揽，独立完成所有的工作。通常，领导者都承担着很多的责任，有很多事情要做，而领导的精力又是有限的，必须将那些下属能做的事情委派给他们，明智地投入到只有领导者才能胜任的活动中去，为群体和组织带来最大的利益。第二，授权使追随者得到了锻炼和成长的机会。下属的成长是领导者重要的责任之一，将重要的任务委派给下属是支持其成长的最好方法。通过授权能够使下属在接受控制和指导的实践中发挥主动性、创造性，从而获得解决问题、创新、管理和决策的技能。授权下的实践是经验积累的过程，也是接受培训的过程和机会。第三，授权使组织系统得到巩固。授权具有激励功能，它向下属传递了这样的信号：他们是被信任的，他们

的成长是非常重要的。有技巧的授权在本质上能提高工作本身的意义和员工的满意度，进而使下属的工作做得更好。同时，一个组织通过完善的授权系统，可以开发员工潜能，使整个组织的管理水平、能力及活力得到提升。

（三）授权中常见的错误

在现实的领导实践中，不少领导者回避授权，这可能是由于以下几种误解：一是认为授权太浪费时间。从短期看，领导者必须花费时间培训下属完成任务，但从长期看，授权能为领导者节约时间。二是认为授权有危险。授权使领导者对工作过程的控制减少，但责任并不减少，这似乎是将领导者自己的声誉押在了其他人的技能和绩效上。其实，衡量和评价领导者是以团队的绩效和成功为依据的，失败团队的领导者不可能得到好的评价。这就要求领导者在委派工作任务时，设置绩效期望和目标，确保任务被下属理解和接受，同时对任务的实施过程和责任状态进行适时、适当的监控。三是认为下属不能胜任工作。通常领导者是组织中最有经验的人，很容易认为下属在技术能力上不如自己，授权会使工作受到损失。其实，即便如此，也没有什么可担心的。领导者应当允许下属在得到授权之初犯一些错误，这是他们为成长必须付出的代价。对一个组织来说，领导者对错误的完美主义式的恐惧可能比错误本身更有害。在授权时，领导者不应该以自己的水平来评估下属，而应该用发展的心态对待下属。四是认为授权会降低领导者的威望。任务是权力和威望的源泉，将重要的任务交给下属，会使领导者失去一定的权力和威望。其实不然，当把重要的和关键性的任务授权给下属时，也增强了下属对领导者的信任和依赖。信任是相互的，只有领导者将重要的任务授权给下属，才能让下属相信领导者是真正信任他们的。另外，通过关键性授权，领导者有机会与下属一起讨论工作目标、任务难度以及实施方案的可行性等问题，避免在重要问题的决策上出现大的失误，这对领导者本身的成长和提高也是非常有利的。

（四）授权的策略与方法

那么，是不是授权越多越好、什么样的授权都好呢？当然不是。授权是一门艺术，需要讲究策略与方法。一般地说，领导者在实施授权时应注意做好以下几个环节：

（1）决定授权的内容。在授权之前，领导者应该对自己目前所承担的或将要承担的所有任务心中有数，评估这些任务所需要花费的时间及其重要和紧迫程度，删除不必要的和益处较小的活动，然后根据情况确定需要授权的活动和任务。

（2）决定授权对象。将合适的工作任务交给适合完成该项工作的下属去做，这是领导授权的一般原则。但是，在具体授权时要注意各个下属或下级群体之间任务和发展机会的平衡性，既不能使工作最出色的部分员工负担过重，即所谓的“鞭打快牛”，也不要使工作胜任力不强的部分员工得不到授权，失去成长和锻炼的机会，造成所谓的“会哭的孩子有奶吃”，领导者要兼顾不同下属的能力、发展与需要来合理确定授权对象。

（3）任务清晰具体。领导者授权时必须确保下属理解任务的内容和具体要求。常见的沟通错误是信息的发出者会高估自己表达的清晰度，领导者在对特定任务进行表达和描述时也常常会如此。一项任务的实施步骤和潜在陷阱对领导者来说可能是显而易见的，但对

下属来说可能并不清楚。因此，授权时领导者要欢迎下属提问题，并对任务和问题做出详细和耐心的说明和解释，必要时还要对关键的注意事项做出书面说明。

（4）指明目标而不是确定程序。领导者在授权时主要是告诉下属要完成什么而不是如何完成。不应该认为自己的方法是最好的，要允许下属使用其他的方法，并鼓励下属提出新观点和进行改革与创新。

（5）对绩效进行监控。有效的授权既不是对下属所做的一切进行事无巨细的管理，也不是对下属放任自流。领导在授予下属一定的工作自主权的同时，还要进行必要的指导与监控，包括在下属遇到难题时的“援助”和下属犯了错误时的纠错、谅解与鼓励。惩罚会挫伤下属创造和革新的动机与热情，尽量不要采用。同时，领导者需要建立具体的程序以便定期评价下属的绩效，提醒和督促下属按照规定的时间和目标要求完成任务。

（6）多赞许，少责备。领导者在把任务交代给下属的同时，也必须将相应的权威一并给予下属，但必须保留对所授权任务需要承担的全部责任和义务，一旦出了问题，领导者应当承担全部的责任，不能推诿。在对下属的绩效提供反馈时，多强调下属做得正确的一面，绩效评价的方法采用所谓的“三明治”法，即“正面评价——负面提醒——正面评价”，至少让下属感到正面与负面的比例平衡，这对于激励下属继续更好地完成被授权的任务是必要的。

二、领导者树立威信的策略与方法

（一）什么是领导者的威信

领导者的威信是指领导者在下属的心理上所受到的肯定、尊敬和信任程度，是下属对领导者的主观评价。威信反应了领导者和下属的心理距离，威信越高，与下属的心理距离越短，反之亦然。威信决定着领导者的影响力和执行力，威信高的领导者能对下属产生强烈的吸引力和感召力，下属更加自愿地服从领导，并自觉按照领导的要求行事。现代管理心理学的研究表明，借助于工资、奖金等激励措施，只能调动员工积极性的60%，其余的40%主要依靠领导者的威信来调动。因此，领导者的威信高低是领导有效性和领导者成败的关键因素之一。

（二）领导者威信的构成

领导者的威信主要由政治威信、道德威信和职业威信三部分组成。领导者的政治威信体现在下属对领导者的政治立场、政治信念和政策水平的信任，这要求领导者具有较高的政治素养和远见卓识，具有较强的政治洞察力和政策敏锐性，在组织面临重大的政策变化和困难时能够立场坚定、明辨是非，为组织和下属确立正确的目标并指明前进方向，并能够给下属以信心和勇气，带领下属共度难关。道德威信是下属对领导者的道德观念、道德修养和道德行为的认可和接受程度。这要求领导者一方面对自己要有严格的道德约束尺度，清正廉洁，洁身自爱，品德高尚，生活作风严谨、正派；另一方面要求领导者在工作上处事公平、公正，以自己高尚的人格魅力感染下属、影响下属和激励下属，发挥道德示范作

用。职业威信是下属对领导者的专业知识、专业技能和领导才能的肯定和信任程度。这要求领导者不但要掌握某一领域的专业知识和技能，而且在处理重大问题和作出重大决策时，能够积极征求和采纳有关方面专家的意见，博采众长，兼收并蓄。虚怀若谷是领导者职业素养的必然要求，也是领导者职业威信的具体体现。实践证明，领导者威信是领导行为有效性坚强的支柱。

（三）领导者的威信感

领导者的威信感是领导者本人对自己威信的主观感受。威信感有真假之分，真实威信感是符合客观实际的威信感，即领导者对自己威信的主观感受与下属对领导者的评价相一致；虚假威信是不符合客观实际的威信感，即领导者的自我感受相背于下属的评价。领导者的虚假威信主要有以下几种形式：（1）压服威信：用权力压服下属，员工强烈不满，但敢怒不敢言，领导者自己反而感到自己很有威信；（2）夸夸其谈威信：对下属空洞说教，言之无物，员工表面认可，背后指责，领导者自我感觉良好；（3）妄自尊大威信：领导者自高自大，盛气凌人，员工表面上忠诚，背后唾骂，领导者觉得自己有威信；（4）收买威信：利用自己的职位权力收买人心、送人情、拉关系、搞帮派，以换取员工的支持，这种交易式的收买具有短期的效果，却得不到员工的长期认可；（5）距离威信：领导者有意与员工保持距离以抬高自己的身份、地位，领导者高高在上或自视清高，不愿或者不敢接近员工，难以和员工形成有效的沟通和情感交流，这样的威信也是虚假的，最终也难以得到员工的认可、理解和支持。

（四）领导者树立威信的策略与方法

领导者的威信归根到底是由其领导素养决定的，但领导者树立正确的威信观，采取适当的策略和方法，也有助于其威信的形成与树立。研究表明，领导者树立威信应注意以下几个方面：

（1）正确对待权力和威信的关系，做到有权有威，相辅相成。权力对领导者威信的积极影响有三个方面：一是传统的看法，即传统上人们习惯认为领导者不同于普通人，有权意味着有才干，这便自然产生下属的服从感；二是职位因素，领导者在组织中的特殊位置会使下属产生敬畏感，这是社会组织赋予领导者的力量；三是资历因素，领导者的学识、生活阅历与经验影响着下属的认知与判断，员工对资深的领导者会产生敬重感，受晕轮效应的作用，资深领导者的言行也容易在员工心理上占有重要位置，易于被接纳。但是，权力对威信的影响是相对的、有限的，过度使用权力、滥用权力会使员工产生逆反心理，损害领导者的威信。

（2）加强自身修养，发挥非权力因素对领导者威信形成的积极作用。影响领导者威信的非权力因素有品格因素、知识因素、能力因素和感情因素。品格因素指领导者的道德品质和作风。领导者优良的品格能够吸引下属并产生榜样力量，甚至会形成员工的精神依赖，而品格不良的领导者会导致员工的反感甚或全盘否定。知识和能力本身就是一种力量，拥有专业知识和解决特殊问题能力的领导者能够产生心理磁场，下属会自觉自愿地跟随并乐于听从领导者的号召，而平庸的领导者是难以得到下属信任的。领导者与下属建立了良好

的情感关系能够使员工产生亲切感。热情、平易近人、和蔼可亲、关心体贴下属的领导者能够创设一种和谐、融洽的群体和组织氛围，给员工带来轻松、愉快和满意的心理感受，这自然就会受到员工的拥护和支持；反之，如果领导者对员工冷漠无情则会导致相互排斥和对抗，既有的威信也会消失殆尽。

（3）正确处理领导者与员工之间水平距与情感距的关系。我国管理心理学家俞文钊教授认为，领导者与员工之间的水平距是指领导者在知识、才能、决策、意志等方面高于员工的程度，感情距是指领导者在情感上与员工的融洽程度，二者与领导者威信之间的关系是：

$$\text{领导者的威信}=\frac{\text{领导者与员工之间的水平}}{\text{领导者与员工之间的感情}}$$

也就是说，领导者的威信与水平距成正比，而与感情距成反比。两种距离交互作用就形成了图 14.1 所示的领导者的四种威信状况：

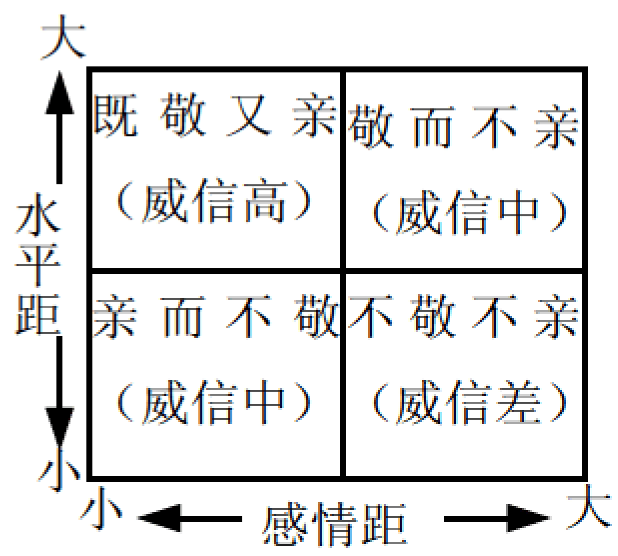


图 14.1　水平距和感情距交互作用形成的四种威信

由此看来，领导者与员工水平距的最大化和感情距的最小化是获得较高的关键所在。那些不学无术而又官气十足的领导者难以在员工心目中树立威信。

三、领导者的会议管理方法

会议是组织运行不可或缺的重要组成部分，组织层级之间的信息沟通与传递、不同工作群体之间的合作与交流很大程度上都是通过会议来完成的。同时，有效的会议还可以促使组织成员之间交流感情，加深彼此的了解与信任，减少猜疑与冷漠，从而增强组织的凝聚力。好的会议能起到事半功倍的效果，但是，不必要的或者无效的会议不但白白浪费了许多时间和金钱，而且令人厌烦和沮丧。一般来说，组织领导者是各类会议的发起者、操纵者和执行者，领导者对于会议的效率和效果起着决定性的作用，同时也是领导者领导能力和水平的具体体现。因此，掌握驾驭会议的艺术，是各级领导的一门必修课。管理心理学家古斯（Guth）和肖（Shaw）提出了召开会议的七点注意事项，对领导者学习如何组织会议是很有帮助的。

（1）确定是否需要召开会议。评估会议召开的必要性，就是要明确会议召开的目的。只有当潜在收益大于潜在成本时，会议才有必要召开。另外，要对会议参加者的状况进行评估，确定他们目前已经掌握的信息、需要了解和接受的新信息以及会议参与者可能的意见等，这将会增加会议的针对性，有助于节约时间和成本。如果是定期召开的例行会议，要注意在每一次会议上安排重要的、实质性的内容，否则就需要对会议进行精简。

（2）列出会议目标。一旦决定有必要召开一次会议，就需要按顺序列出会议的目标并制定一项实现这些目标的计划。要估计每一项会议议程安排所需要的时间，如果能事先将会议议程和需要解决问题告诉与会者，让他们提前进行必要的准备，效果会更好。

（3）按议程表进行。会议一旦开始，领导者就坚持按照事先制定的议程表进行会议流程。群体很容易转移目标，离题是许多会议常常存在的弊端。如果对于一些较困难的决策问题一时难以做出结论，可以暂时搁置或者约定专门的时间再行商议，不可因为一件事情影响整个会议进程。有效的会议切忌冗长、空洞和“马拉松”式的议而不决。

（4）事先提供相关资料。调查表明，许多效率低下的会议都是把时间耗费在阅读会议期间派发的材料和等待与会者对这些材料进行理解和思考上。如果能将部分事先准备好的会议材料提前分发给与会的有关人员阅读，既可以节约会议时间，提高会议效率，又可以提高会议决策质量和效果，还可以减少因信息沟通不及时带来的不满情绪。

（5）使会议尽可能地方便。应尽可能挑选对所有与会者都很方便的时间和地点召开会议。这不但可以提高会议出勤率、减少其他事务对会议的干扰，还有助于聚焦会议的关键性问题和集中精力解决难题。一般情况下，会议要准时开始和按规定的时间结束，等待迟到者是对准时到达者的不公平，还会传达会议无关紧要的错误信息，领导者作为会议的组织者和主导者要有强烈的时间观念，坚持并自觉遵守会议时间表。

（6）鼓励参与。通常而言，与会的每一个人都应当有发言的机会，并对会议的成果有所贡献，这是与会成员对群体做出贡献的机会，也是会议的价值体现。领导者要采取积极的方式鼓励不爱发言的成员说出他们的意见和看法，避免沉默式的赞同和同意，同时也消除了“会上同意会后反对”现象的发生，有时还需要缩短那些言之无物而又乐于发言的与会者的表述时间。创造宽松、民主、和谐的会议氛围不失为一种有效的鼓励方式，应让与会者做到知无不言、言无不尽，只有这样，才能真正地体现会议的功能和作用。

（7）做好会议纪录。做好会议纪录是成功会议不可缺少的环节。一方面，把会议所讨论的观点、各项决策以及所采取的行动完整地记录下来，有助于领导者会后对这些问题进行再思考，有时与会者的观点对领导者会有新的启发；另一方面，会议记录有助于落实会上做出的各项决策和监督检查执行情况，包括明确谁应对那项任务负责以及完成任务的期限等。同时，会议记录还能为未来的会议提供有益的参考信息，以及将来对有关问题及责任的备案与处理等。简要的记录可以由领导者自己完成，重要的会议需要指定专门人员做好会议记录并对会议纪录做好存档工作。

总之，按照以上要求举行各类会议，能帮助领导者更加有效地实施领导职能，也能更好地规范追随者的行为，促进组织目标更好地完成。

第二节 领导者人力资源开发的策略与方法

人力资源开发（HRD）正在逐步取代人力资源管理（HRM）而成为组织管理实践的重心。之所以如此，一方面是因为在崇尚人性化管理的今天，人在组织中的成长和发展越来越受到重视，逐渐成为企业人力资源管理要实现的目标之一。人力资源开发是实现这个目标的重要途径。另一方面是因为，在不断变化的竞争环境中，企业的发展或组织目标的实现需要不断提升员工的知识、技能水平，更新员工的观念，塑造员工的态度。越来越多的企业认识到，与其不断地招聘和引进新员工，不如就地开发和挖掘现有员工的潜能和价值，而要做到这一点，仅依靠过去的人力资源管理手段如岗位分析、招聘、培训、考评、薪酬等是远远不够的。下面从几个方面分析领导者人力资源开发的策略与方法。

一、员工职业生涯管理的策略与方法

（一）组织职业生涯管理的概念和意义

组织职业生涯管理（organizational career management ,简称 OCM）是指由组织实施的、旨在开发员工的潜力、留住员工、使员工能自我实现的一系列管理方法。许多企业如 IBM 、Xerox、Hewlett-Packet、Disney 等都相继在企业中实施了组织职业生涯管理，而且取得了比较理想的效果，成为许多企业纷纷仿效的榜样。

组织职业生涯管理对组织和个人的发展都具有十分重要的意义。从组织的角度看，首先，职业生涯管理是企业人力资源合理配置的首要问题。人力资源是一种增量资源，特别是随着知识经济时代的到来，知识已成为企业获取竞争优势的重要资本，而掌握和创造这些知识的是“人”，因此企业更应注重人的智慧、技艺、能力的提高与全面发展。其次，职业生涯管理具有长效的激励功能，虽然其成效不如薪酬那么及时，但它能充分调动人的内在积极性且作用比较持久，可以留住员工的心，使员工具备持续发展的潜力。从个人的角度看，首先，组织职业生涯管理为员工更好地认识自己提供了机会。职业计划和职业管理既能使员工了解自身的长处和短处，养成对环境和工作目标进行分析的习惯，又可以使员工合理计划、分配时间和精力完成任务、提高技能，从而在不断认识自我的过程中增加自身的竞争力。其次，它有利于员工过好职业生活，实现职业目标。职业生涯管理可以帮助员工从更高的角度看待工作中的各种问题，将分离的事件联系起来，从而使职业生活更加充实和富有成效。它还有助于员工平衡职业生活同个人追求、家庭目标等之间的关系，避免顾此失彼的困境。第三，职业生涯管理能满足个人的归属需要、尊重需要和自我实现需要，使个体实现自我价值的不断提升和超越。工作的最初目的可能仅仅是找一份养家糊口的差事，进而追求的可能是财富、地位和名望。职业生涯管理可以帮助员工实现职业目标的不断提升，使工作目的超越财富和地位之上，追求更高层次的自我价值实现。

（二）组织职业生涯管理的原则和步骤

1．组织职业生涯管理的原则

（1）可行性原则：规划设计要以员工自己的实际情况和组织现实为依据，设计的方案要可行，一切主观虚构的美好幻想和不着边际的梦想都是没有意义的。

（2）适时性原则：规划是预测未来的行动，在确定职业目标时，要有时间上和时序上的妥善安排，对行动的实施和完成时间要有具体的描述，以作为检查行为的依据。

（3）适应性原则：设计未来的职业生涯目标，涉及到诸多方面的可变因素，因此规划要有适度的弹性，以增加其适应性。

（4）持续性原则：职业生涯规划的每一个阶段要连贯衔接、循序渐进，不可跨越或跳跃。

2．员工的职业生涯管理的步骤

（1）综合评估与职业定位：通过对员工特点及环境的综合分析，对其能力、兴趣、潜力、职业生涯需求及追求目标作出评估，帮助员工选择和设计他们的职业道路。其内容主要包括对员工的职业价值观、职业能力、职业锚以及职业发展潜能进行评估，常用的方法有职业性向测验、橱窗分析法以及能力测验、心理测验等。它能使员工在职业生涯规划中充分考虑个人责任，检查过去的职业选择效果，鼓舞员工尝试新的工作，挑战更加艰巨的任务。在这个过程中，员工深刻的自我剖析也是非常重要的。

（2）职业生涯发展机会评估：首先要对员工所处的社会环境、行业环境、组织环境及目前职业状况进行分析，在此基础上，评估各种环境因素对其职业发展的影响，根据员工的兴趣、爱好与特长，结合其性格、气质与能力等特征，对其职业发展的各种机会进行评估。SWOT 分析技术是广泛使用的评估工具，即通过员工自身因素与社会因素的比较，寻找优势、劣势、机会与威胁。

（3）职业生涯目标设定：在对员工的综合素质和发展机会充分认知的基础上，对其职业生涯的短期目标、中期目标和长期目标进行规划。目标设定一般要遵循以下原则：第一，目标的明确性，即要明确描述个人在完成每一目标时的行动方案；第二，目标的可测量性，目标要有定量的数据指标，便于检查；第三，目标的时限性，要设定目标的特定完成时间；第四，目标的可实现性，目标要是通过一定努力可以实现的。

（4）制定职业生涯规划实施方案：在帮助员工制定好职业生涯目标之后，还要制定切实可行的实施方案，督促员工严格按照计划行事，保证方案最大限度得到贯彻执行，当然也包括为员工提供条件、打通路径、扫除障碍等配套组织措施。

（5）职业生涯规划调整：员工在实施职业生涯方案时，可能会遇到这样那样的困难或挫折，所以需要经常地给予帮助和支持，最重要的是帮助他们掌握应付困难和挫折的策略和方法，如自觉地总结经验和教训、评估规划和调整认知、纠正目标偏差和修改行动路线、变更策略计划等。具体而言，可以采用以下几种策略：①经常回顾自己的构想和行动规划；②如果理想蓝图发生变化，行动规划要及时做出适当调整；③常与朋友和领导讨论构想和行动方案；④抓住组织提供的可能机会；⑤每三个月检查工作进度；⑥学会拒绝应酬，增加精力投入等。

（三）员工职业生涯管理的策略与方法

格特瑞杰（Gutteridge）概括了西方组织职业生涯管理的方法，具体包括：给个人提供自我评估的工具和机会；进行个别职业发展咨询；发布内部劳动力市场信息；设置潜能评价中心；实施培训、发展项目等。

领导者在员工职业生涯开发、规划和发展过程中发挥着重要的指导作用。一方面要对员工的需要和发展能力做出总体判断，另一方面还要组织各项管理制度和人事制度的制定，提出未来组织人力资源发展的战略规划。一般情况下，组织最高领导者较少、也不应该参与个体员工职业生涯的指导，通常只应该处理有潜能的员工和组织高层员工的职业生涯发展问题。但是，人力资源管理者和员工的直接上级在员工职业生涯管理上扮演着重要的角色，其作用主要体现在：（1）对员工职业潜能定位；（2）通过不同工作任务分派来促使员工发挥自身潜能；（3）根据对员工的评估对其职业生涯提供指导，充当顾问角色；（4）利用权力促进员工在组织内的职业发展和晋升。可以说，每一个组织领导者都从不同层次和侧面影响着员工职业生涯规划和发展，组织应建立员工职业生涯管理的工作体系，以更好地促进组织和员工的共同发展。

二、员工培训的策略与方法

（一）员工培训的含义及功能

员工培训是为了实现组织目标而进行的提高员工知识、技能和能力的过程。狭义的培训是针对特定岗位的知识和技能培训，广义的培训还包括能力开发、观念更新等范畴，强调获得对目前和将来具有适应性的能力或观念，侧重于综合能力的培养和开发。

组织的领导者们越来越深刻地认识到了员工培训的重要性。许多组织把它列为人力资源开发战略的首要任务。资料表明，美国企业每年花费在培训上的费用至少在 50 亿美元以上，其中 2/3 的培训费用于开发职业经理人员，1/3 的费用用于培训一线工人。特别是当组织进行重构和发生战略转变时，培训就显得更加重要。员工要适应变化的形势就必须通过接受培训来更新知识、技能和观念。同样地，管理者也需要参加培训以增强他们的决策能力、指挥能力和领导水平。培训不仅可以带来生产率的提高，还能够形成新的生产力，同时好的培训有助于增强组织凝聚力和向心力，增强组织的竞争力。

（二）培训的种类或形式

从培训的内容来看，目前培训的内容十分广泛，几乎涉及到组织和员工个人的各个方面，如新员工适应、管理绩效评估、电脑操作、团队建设、领导才能、性骚扰应对、新设备操作、人才选拔过程以及对培训者的培训等。

从培训的方式上看，可分为组织内部培训和外部培训两种形式。内部培训有：（1）岗位技能培训，即针对员工特定工作岗位的操作技能培训。这种培训节省了派出员工接受培训的费用开支，但如果培训者素质不过硬或培训条件不具备的话，常常达不到预期效果，容易在后期的工作中产生较高成本（如顾客流失和设备损坏），同时这种培训通常不具有激

励员工的功能，因为员工一般不把这种培训理解为一种福利或发展机会。（2）技术培训：通常是针对更新的通用技术（如新的技术设备、新开发的管理软件、新的绩效考核方法等）的使用问题，在组织内部抽出专门时间，将需要培训的员工集中起来，由专门人员进行培训。（3）非正式培训，这种培训是通过内部员工之间的互相影响和反馈实现的，即员工通过观察、模仿或向其他员工请教等非正式途径来获得技能、观念、态度，以适应工作。外部培训是在内部培训资源不足的情况下组织所采取的培训方式。通常有两种方式：（1）依靠产品供应商培训；（2）依靠社会专门培训机构培训。许多企业利用供应商来培训自己的员工，如软件供应商举行用户会议、电脑供应商为商户提供技术资格证书培训等，这种培训对组织来讲花费较低。有时，组织根据需要派员工到社会上的专门培训机构或高等院校接受培训，如会计师资格证培训、审计师资格证培训、MBA 学历培训等。资格证培训和学历培训是组织选拔、培养和激励员工的有效方式，有利于培养组织核心员工，增加员工对组织的忠诚感，起到稳定和留住优秀人才的作用。

（三）员工培训的策略与方法

实践证明，领导者是否重视和积极参与员工培训直接关系到培训的效果，并间接地影响组织的绩效和发展。有远见的领导者都能够将员工培训作为人力资源开发的首选途径来看待，从来不会因为吝惜培训成本而忽视或放弃培训。无数成功领导者的经验表明，在员工培训问题上，领导者应该、也能够做到以下几点：首先，领导者要有学习型组织的观念，教育和引导员工积极参与培训，鼓励员工不断提高自身能力水平，更新观念，追求自我成长和发展；第二，领导者要指导人力资源管理部门建立员工内训和外训管理体制和实施计划，做到规范化、制度化和系统化；第三，领导者要成为积极参与培训的表率，起到示范带头作用；第四，领导者要把员工培训与对员工的绩效考核、晋升等结合起来。

三、员工“工作—家庭”平衡的策略与方法

（一）员工的工作—家庭冲突

任何一个员工在生活中都必然会遇到工作与家庭的矛盾或冲突问题，如何在二者之间取得平衡，化解二者之间的冲突和矛盾，是员工面临的难题之一。而这个难题解决得如何直接影响员工的工作态度、工作表现、绩效水平和职业前景，进而影响组织的发展，所以员工的工作—家庭平衡问题也是组织人力资源开发面临的重要课题之一。

实际上，除了工作和家庭之外，员工还有自己的个人生活空间和自我发展问题。图 14.2 给出了员工生活的三个区域及其相互之间的冲突模型。每一个员工都需要将一部分时间和精力投入到自我发展区（A 区），如自己的业余爱好、社团生活、友谊等，也需要将一部分时间和精力投入到工作区（B 区），还需要把一部分时间投入到家庭生活区（C 区）。这三者之间并非相互独立、互不影响，而是经常会相互挤占对方的空间，产生冲突与矛盾。

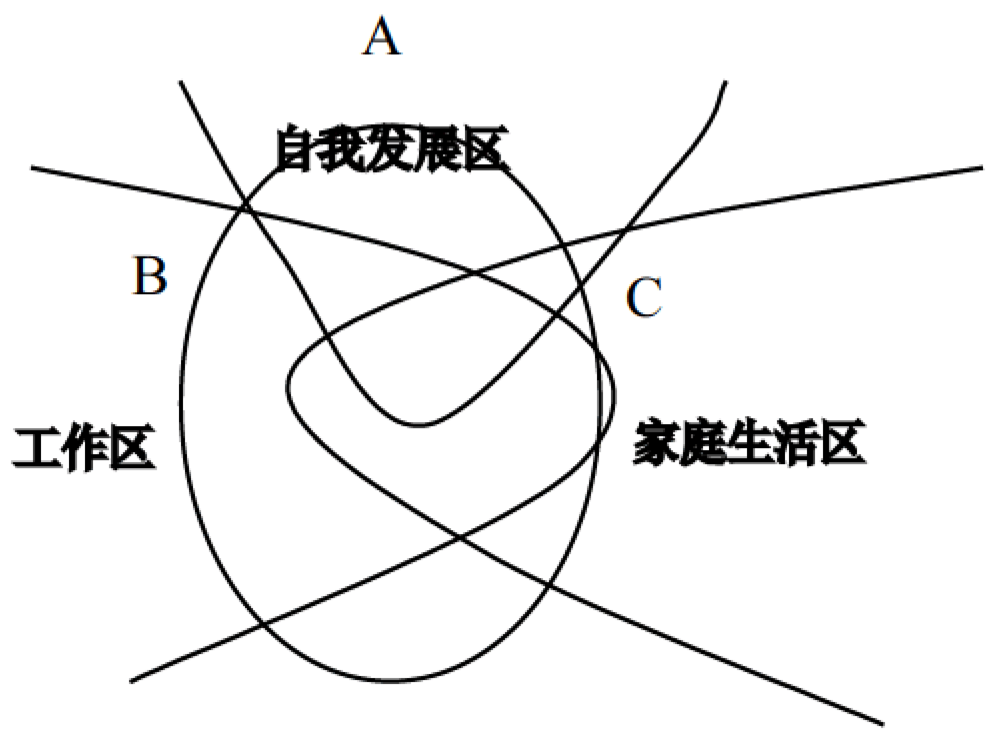


图 14.2　自我发展、工作和家庭之间的冲突模型

研究表明，员工的工作—家庭冲突主要表现在以下几个方面：（1）工作占用了与家庭成员相聚的时间和承担家务的时间；（2）工作地点与家庭所在地之间的矛盾，也就是说员工所喜欢的家庭居住地很可能不是工作单位的所在地；（3）在工作上的收入水平、职位高低及职业声望与在家庭中的角色地位之间的矛盾；（4）围绕工作的感情气氛与家庭成员的感情气氛之间的相互影响。这四个方面的关系如果处理不当，常常导致各类冲突的发生，如时间冲突、情绪冲突、行为冲突和角色冲突等，这都将会给工作和员工个人发展带来不利的影响。

（二）工作—家庭平衡的理论基础

关于工作—家庭关系的研究非常活跃，早期较有影响的理论是“溢出理论”和“补偿理论”。后来，美国学者克拉克在批判性地总结了前人研究的基础上，提出了“工作—家庭边界理论”。该理论认为，工作和家庭属于两个不同性质的生活领域，二者之间不存在天然的联系，却存在明显的边界。将工作与家庭联系起来的不是感情，而是人。人们每天在工作和家庭的边界上徘徊，被迫不断地在工作和家庭两个领域间穿梭转移，同时塑造着两个领域的边界，边界影响着不同领域内的成员关系。在领域穿行的过程中，人们不断调整焦点目标及个人的“内心风格”以适应每个领域独特的要求。边界理论告诉我们，虽然工作和家庭之间的冲突难以调和，但个体可以通过调整工作与家庭的边界来达到二者的平衡。平衡是实现工作和家庭生活满意度和减少工作与职业角色冲突的桥梁。该理论既描述了工作—家庭冲突的原因，又给出了解决工作—家庭冲突问题的框架，为组织和员工更好地处理工作与家庭关系提供了工具性支持。

（三）工作—家庭平衡计划

工作—家庭平衡计划是组织为帮助员工协调和平衡工作—家庭冲突而建立的社会支持系统和心理保健系统，是组织人力资源开发计划的重要组成部分。一般地说，该计划的实施需要做好以下几项工作：

（1）加强与员工的信息沟通，及时了解和切实关心员工的家庭困难，有针对性地给予

帮助，如提供弹性工作制、建立婴儿与老人照料中心等。

（2）向员工提供家庭问题和压力排解的咨询服务。如设立专门机构，选配专业人员，向员工提供家庭问题或心理问题的咨询服务。例如，第十一章详细介绍过的员工援助计划就是解决员工工作—家庭平衡问题的重要途径之一。

（3）实施职务分担制度，即由两名或多名员工共同承担某项全职工作，以便员工有时间照料家庭。

（4）组织支持性服务，如对员工进行培训、灵活的福利计划等。

第三节　领导者处理人际关系的策略与方法

一、领导者人际关系概述

（一）人际关系的含义、特点及影响因素

人际关系是人们在共同的社会活动过程中彼此之间心理上的距离与联系，是社会交往的结果。它既包括心理关系，也包括行为关系。不同的人际关系会引起不同的情感体验。人与人之间如果能够满足对方的社会需求，则能够形成与保持接近的心理关系，表现为相互友好、相互喜爱的情感，双方就会感到心情舒畅，心理距离就会越来越亲近，反之，双方关系就会疏远或紧张。

相对于其他关系而言，人际关系有其自身的特点。首先，社会性是人际关系的根本特征，是把人类群体关系同动物群体关系区别开来的特征。人是社会的人，一个人的生存和发展离不开与别人的交往和联系。美国心理学家梅奥通过霍桑实验，提出了人际关系学说，认为人除了物质、经济的需求以外，还有社会的、心理的需求，这些需求是通过人际交往来满足和实现的。社会性的另一个表现是人们在交往过程中必须遵循一定的社会规范，包括一定社会历史条件下形成的法律规范和道德规范。离开或违背了这些规范，难以形成良好的人际关系。其次，人际关系具有情感性特征。人是富有感情的动物，人们之间的交往是带有感情色彩的，感情因素维系着人际互动，同时人际交往的目的之一就是满足人们的情感需求。感情是人与人交往的基础，是亲密人际关系建立的基本条件。第三，人际关系具有复杂性。在人们的交往过程中，各种关系交织在一起，错综复杂。心理的、行为的、感情的、利益的以及政治、经济、伦理等各种因素共同制约和影响着人际关系。

人际关系的亲近与疏远表现为人们彼此之间吸引力的大小。研究表明，影响人际吸引的因素主要有：（1）临近性和熟悉性。俗话说：近水楼台先得月，向阳花木早逢春。一般情况下，交往频率越高，相互之间越熟悉，就越有可能彼此吸引。地理位置上临近，相互接触机会就增多，也就更容易了解对方。（2）相似性与互补性。在社会交往中，人们对于那些与自己具有某些相似性特征的人心存好感，特别是在年龄、职业、社会地位、文化水

平、经历以及价值、态度等方面与自己相似的人，容易产生好感。互补性是指当一方具有的品质正好是另一方所不具备的时候，前者对后者就会有吸引力。（3）外貌。个人的长相、衣着打扮、姿态等都会对彼此的吸引力产生一定影响，尤其在双方第一次接触的时候，由于第一印象的作用，人的外表对双方关系的发展具有重要作用。（4）对等性。人们总是希望自己被他人所接受，因此更乐意与喜欢自己、欣赏自己的人交往。（5）能力。一般情况下，能力越强的人越受人欢迎。另外，人们的人格特征、双方的默契程度等也影响人们之间的相互吸引。

（二）领导者的人际行为模式及其重要性

领导者的人际行为模式是影响领导有效性和领导者影响力的重要因素之一。领导者处理人际关系的不同方式会引起员工不同的感受和反应，从而影响领导效果。心理学家李雷（T.F.Leary）总结概括出了如表 14.1 所示的八种领导者人际行为模式及其后果。

表 14.1　领导者八种人际行为模式及其不同后果

领导行为模式	下属的反应
管理、指挥、劝告、教育	尊重和服从
关心、帮助、支持	信任和接受
合作、同意、友好	协助和温和
尊敬、信任、赞扬、请求帮助	劝导和帮助
害羞、礼貌、服从	骄傲或控制
反抗、疲倦、怀疑	惩罚或拒绝
攻击、惩罚、不友好	敌对和反抗
激烈、拒绝、夸大、炫耀	不信任或自卑

显而易见，领导者的人际关系比一般的人际关系更为复杂。领导者以放射型的方式同上级、同事、下级和其他群体发生着全方位的人际交往。每一位领导者都时刻面临着如何处理好这些关系的问题。无论哪方面的关系不协调，都很难顺利开展工作。即使领导者与下属关系融洽，领导者群体也很团结，但如果与上级关系不好，仍将一事无成，最终可能被调离或者被降职；如果领导者与上级、下级的关系很好，但领导者群体内部不团结，人际关系不融洽，工作中不能劲往一处使，形不成有效的合力，也难以把工作做好；如果领导者与下属关系不好，领导者的意图不能很好地被下级贯彻落实，难以有效地调动下属的工作热情与积极性，不仅难以完成工作任务，还会导致下属的抵制、对抗情绪和行为，久而久之，就会使组织处于相当被动和不利的局面。

二、领导者处理与上级关系的策略与方法

对于处在组织中特定职位的领导者来说，上级就是比自己拥有更多权力和权威的人。领导者处理好与上级的关系无论对组织发展，还是对领导者个人的发展，都是非常重要的。

从组织发展的角度看，良好的上、下级关系，共同的价值观、态度和方法，可以减少矛盾和冲突，相互提供更多的支持，这有利于工作的顺利开展和目标的如期实现。从领导者个人发展的角度来看，拥有良好的上下级关系，有利于得到上级更多的提携，在决策中拥有更多的发言权，并获得一些重要的工作任务，在职业生涯上得到支持。相反，如果与上级的关系不好，不仅难以得到有效的支持，甚至难以得到公正的评价，当然也不会有好的前途。

在处理与上级的关系时，很多领导者往往有一些误解，认为自己与上级的关系纯粹是运气决定的，自己处于被动地位，是否被上级喜欢和器重不是由自己决定的。其实不然，上下级关系的质量并非完全取决于上级，下属应学会采取积极的步骤来加强与上级的联系。实际上，不管个人处于组织的何种职位，都有可能帮助上级取得成功。领导者处理与上级的关系主要应该注意以下两个方面：

（1）理解上级的目标和意图。首先，在了解了上级的价值观、偏好和个性的基础上，弄清楚上级做事情的真正目标和意图，这有助于下级更好地贯彻执行上级的指示，增强双方的关系。其次，下级也要意识到，上级不是超人，他们也有弱点，在贯彻执行上级目标和意图过程中，方法上要有创意和创新，弥补上级的弱点和不足，这不仅可以得到上级的赞赏，更重要的是有助于提高和改善组织绩效。第三，设法使上级更多地了解自己，多征求上级的意见和建议，强化上下级关系。很少有上级不喜欢倾听下级汇报、了解下级工作情况和对下级提出意见和建议的。

（2）主动适应上级的领导风格。研究表明，一些下级领导者得不到上级认可或器重，是因为他们没有能力或者不愿意适应上级与自己不同的领导风格。下级领导者应当牢记，适应上级的领导风格是自己的责任。不同的领导者有不同的领导风格。例如，有的领导者雷厉风行，有的领导者做事缜密；有的领导者只管结果不问过程，而有的领导者喜欢事无巨细；有的领导者喜欢听下级的口头汇报，有的领导者则喜欢书面汇报和下达书面指示等。因此，作为下级需要灵活地适应上级的决策风格、问题解决策略、沟通方式和互动策略等。此外，有些领导者会主动地讲清楚他们对下级的角色要求，但大多数领导者不愿意或讲不清楚对下属的角色要求，他们期望下属能够明白自己应该做什么、怎么做。下级可以主动地列出自己的主要工作职责，然后与上级进行沟通和交流，征求上级领导的意见和建议，这样也可以加强与上级领导的关系。

三、领导者处理与下属关系的策略与方法

（一）领导者与下属关系的重要性

领导者与下属的关系既是决定领导者影响力的前提条件，也是影响领导行为有效性的重要因素。领导就是领导者与下属不断产生互动的过程，领导者最重要的职能就是激励下属完成目标和提高下属的工作满意度。能否做到这一点，取决于领导者与下属互动的效果，即互动所导致的领导者与下属的人际关系。成功的领导者总是能够把各方面的专门人才团

结在自己周围，让他们心情舒畅地工作，以集体的力量来建功立业。美国钢铁大王卡内基的墓志铭上写着这样一句话："这里长眠着一位善于同高于自己的人一道工作的强人。"这句话一言道破了领导者与下属关系的重要性。

（二）领导者处理与下属关系的原则

在处理与下属的关系时，领导者应遵循以下原则：

（1）以身作则原则：要求下属做到的事情领导者首先要做到，甚至做得更好，以起到示范作用。

（2）尊重下属人格原则：尽管在组织关系中领导者与下属有上下级之分，但在人格上是平等的主体关系，下属的人格地位是独立的，领导者应予以足够的尊重。

（3）相互利益原则：领导者主动关心下属的利益，并尽可能满足下属的需要，使下属产生敬重和服从感。

（4）积极激励原则：领导者要不断地开放和持续地激励下属的工作积极性，做到持之以恒。

（5）沟通授权原则：领导者要通过沟通网络，经常与下属进行交流，增加彼此的了解，还要及时有效地进行授权，营造互助与合作的氛围。

（6）民主参与原则：领导者要尽量扩大下属民主参与的机会，用"权力平衡"来代替"简单的命令服从"，增强下属的责任意识、主动意识以及对领导者的信任和依赖。

（三）领导者缩短与下属心理距离的策略与方法

我国管理心理学家俞文钊教授认为，领导者要想缩短与下属的心理距离，融洽与下属的人际关系，需要做到以下几点。

1．深入了解下属的行为动机

从理论上讲，每一个工作群体都有一个共同的目标，全体员工都应该有共同的认识，但事实上，由于每个人的人生观和价值观不同、经历和教育程度不同、所处的社会地位不同以及考虑问题的角度和重点不同，下属的想法和领导者的想法不可能完全一致，认识和动机也是千差万别。因此，领导者要深入了解下属的不同想法，只有这样，才能有助于双方达成共识，最终实现个人目标与组织目标的统一。

要认识下属的行为动机，领导者一方面要客观地搜集下属对自己的反应，虚心听取不同意见，以纠正自己的片面性，另一方面要学会经常检视自己内心深处真正的动机与情感，进而设身处地地理解和把握下属的真实动机与想法。

2．重视下属的尊重需要

每一人都有自尊的需要，领导者对待下属，要特别注意方式方法，不能使他们失去面子。实际上，下属为了维护领导的面子，总是千方百计地去满足领导者和同事的要求，如果领导者不考虑下属的面子，就容易使下属格外伤心，从而丧失前进的勇气，有的甚至会自暴自弃或反目成仇。因此，不伤害并千方百计地保护下属的自尊心，是领导者处理与下属关系的第一要务。

3．正确对待下属的过失行为

当下属有过失行为时，领导者既要留有情面，又要不失时机地予以处理，做到仁至义尽和赏罚分明并重。具体而言，应注意以下几点：

（1）不当众责骂下属。

（2）与下属谈话时，不带蔑视的口气和情绪。

（3）不扩大下属的错误和过失，不上纲上线。

（4）以平等的态度对待下属。

（5）尽可能体谅和容忍下属的过失，给下属创造改正错误的机会。

对于有过错的下属，领导者如果处理得当，会使下属和领导的关系更加亲近，否则会使下属疏远自己，甚至离心离德，有的会成为领导者的对立面，带来不利的影响。

本章小结：

1．授权就是领导者向下属下放权力和责任的一种管理方法。授权将领导者从烦琐的杂务中解放出来、使追随者得到了锻炼和成长机会、还能够起到激励员工的作用。领导者授权时应注意授权的内容和对象，做到任务清晰具体、指明目标而不是确定程序、对绩效进行监控、多赞许，少责备。

2．领导者的威信是领导者在下属心理上所受到的肯定、尊敬和信任程度，是下属对领导者的主观评价。它主要由政治威信、道德威信和职业威信三部分组成。领导者的虚假威信主要有压服威信、夸夸其谈威信、妄自尊大威信、收买威信、距离威信等几种形式。领导者树立威信应注意做到正确对待权力和威信的关系、加强自身修养、正确处理领导者与员工之间水平距与情感距的关系。

3．会议是组织运行不可或缺的重要组成部分。领导者召开会议前后应注意确定是否需要召开会议、列出会议目标、按议程表进行、事先提供相关资料、使会议尽可能地方便、鼓励参与、做好会议纪录。

4．组织职业生涯管理是指由组织实施的、旨在开发员工的潜力、留住员工、使员工能自我实现的一系列管理方法。它对组织和个人的发展都具有十分重要的意义。组织职业生涯管理应遵循可行性原则、适时性原则、适应性原则、持续性原则等。职业生涯管理的步骤包括综合评估与职业定位、职业生涯发展机会评估、职业生涯目标设定、制定职业生涯规划实施方案、职业生涯规划调整等。组织可以采取给个人提供自我评估的工具和机会、进行个别职业发展咨询、发布内部劳动力市场信息、设置潜能评价中心、实施培训等措施管理员工的职业生涯。

5．员工培训是为了实现组织目标而进行的提高员工知识、技能和能力的过程。培训的内容十分广泛，培训的方式主要有组织内部培训和外部培训两种形式。领导者应该树立学习型组织的观念、建立员工内训和外训管理体制和实施计划、成为积极参与培训的表率、把员工培训与对员工的绩效考核、晋升等结合起来。

6．工作—家庭冲突是员工面临的难题之一，也是组织人力资源开发面临的重要课题之

一。工作—家庭平衡计划是解决这个问题的有效途径，它是组织为帮助员工协调和平衡工作—家庭冲突而建立的社会支持系统和心理保健系统。具体措施有及时了解和切实关心员工的家庭困难、向员工提供家庭问题和压力排解的咨询服务、实施职务分担制度、提供支持性服务等。

7．人际关系是人们在共同的社会活动过程中彼此之间心理上的距离与联系。它具有社会性、情感性、复杂性等特征。临近性和熟悉性、相似性与互补性、外貌、对等性、能力等影响人们之间的相互吸引。领导者的人际行为模式是影响领导有效性和领导者影响力的重要因素之一。领导者处理与上级的关系时应注意理解上级的目标和意图，主动适应上级的领导风格。领导者处理与下属的关系时应遵循以身作则原则、尊重下属人格原则、相互利益原则、积极激励原则、沟通授权原则、民主参与原则，并做到深入了解下属的行为动机、重视下属的尊重需要、正确对待下属的过失行为。

本章思考题：

1．领导者有效授权的意义何在？如何进行有效授权？
2．领导者如何建立真威信、避免假威信？
3．领导者如何提高会议的有效性、增强针对性？
4．组织实施工作—家庭平衡计划的依据、意义何在？如何实施？
5．结合组织管理实际，谈谈如何进行员工职业生涯管理？
6．领导者如何正确处理与上级、下级的关系？

案例分析：

The parlor 是旧金山的一家经营三明治和小型晚餐的营业店，它主要的吸引力在于用高品质的老式苏打泉水制成的优良的圣代冰淇淋和汽水，优质的三明治也颇受食客的青睐，开业七年来，生意一直稳步增长。

随着经营规模的不断扩大，The parlor 经营店的所有者、管理者理查德 · 帕维斯决定雇佣一名营业经理，以便使自己把更多的时间和精力投入到其他更重要的工作中。经过一个月的招聘和面试，他选择了保罗 · 麦卡锡，麦卡锡曾经做过镇上久负盛名的滑稽表演剧场的点心、饮料部门的管理者。

麦卡锡第一天上班是在没有人陪同的情况下走入营业店的（当时帕维斯不在镇上），他介绍了自己，雇员们才得知他的身份。

在开始的几个星期，麦卡锡很负责地完成他的管理工作，并看起来很有效率，根据合同，他的报酬是固定薪水加上为店里各项节约费用的百分比提成，别的雇员的报酬是按照固定的小时工资支付的。

一个月后，麦卡锡全身心的投入到节约计划之中，这样可以提高他的收入。他武断地做出了许多决定，如更换原料供应商、减少菜单上每种菜的份额、延长低工资率雇员的工作时间、减少高工资率雇员的工作时间等，当雇员们对麦卡锡的新举措提出质疑时，他发

誓说这些举措对公司没有负面影响。于是一批老雇员纷纷离开，只能重新招聘新人。

罗恩·夏普是附近大学的会计系学生，在麦卡锡来这里之前曾做了 5 个月的夜班快餐厨师，工作尽职尽责，工作纪录一直保持良好。一天，罗恩的一名同事（负责白班的厨师），因迟到 5 分钟且与麦卡锡大吵，麦卡锡立刻解雇了他，因此，在罗恩凌晨 3 点下了夜班之后，麦卡锡要求他顶替那位被解雇的厨师在早晨 9 点继续上班（按规定罗恩通常上班的时间是下午 4 点）。但是，当罗恩困倦不堪地在上午 10 点钟到达时（上了一节 8 点钟的会计课），麦卡锡大发脾气，暴跳如雷，他警告罗恩，“再有一次你就可以去找另一份工作了。如果你为我工作的话，就要按照我的方式做事，否则你就别在这里工作。”“这对我来说真是好极了！”罗恩立刻回击了过去，愤然地将围裙扔进了水池里。

问题：

1．帕维斯的授权得当吗？

2．麦卡锡的领导风格、领导艺术到底出了什么问题？

第十五章　胜任特征模型及领导干部选拔

前面两章对领导和领导者的概念、领导有效性理论及领导策略和方法等问题进行了探讨。这只是回答了怎样让已有的领导者做得更好的问题，却没有回答如何科学地选拔和培养领导者的问题。实际上，后一个问题才是许多组织和企业更加关心的问题。本章将从胜任特征模型入手，引申出领导者胜任特征模型的概念，然后对近年来出现的、应用较多的领导者评价和选拔方法加以介绍。

第一节　胜任特征模型

一、胜任特征的含义

美国哈佛大学著名心理学家麦克莱兰（McClelland）于 1973 年发表了一篇题为“测量胜任特征而不是智力”的文章，明确提出了胜任特征（competency）的概念，并指出胜任特征是指那些能够区分在特定工作岗位和组织环境中绩效水平的个人特征。这一概念使当代的人才观发生了巨大转变，由单纯以智力高低为标准转变为以个人综合素质为标准，将人员的素质特征与实际工作岗位特点直接联系起来，突出了实际工作中解决问题的能力（许多拥有高智商的个体不能胜任某些实际工作）。

胜任特征概念一经提出，便迅速成为研究的热点。经过 20 多年的发展，在大量研究基础上，斯本塞（Spencer）于 1993 年重新对胜任特征进行了界定：胜任特征是指和参照效标有因果关联的个体的潜在特征，或者说是能将某一工作岗位上表现优秀者和表现一般者区分开来的个体潜在特征。道尔顿（Dalton，1997）认为，胜任特征就是那些让个体在工作中脱颖而出的实际行为。杜博斯（DuBois，1999）认为，胜任特征是指那些在实际工作中有效完成任务所必需的特征或属性，具体来讲，就是那些表现优异者所拥有的特征。虽然上述几种表述略有差异，但本质是一致的：首先，胜任特征是指一组个体拥有的特征；其次，这组特征与高绩效（表现优秀）具有因果关系，或者说是导致高绩效的直接原因。

二、胜任特征模型的内容或结构

胜任特征模型是指担任某一特定任务角色所需要具备的胜任特征的总和，是一种包含多种胜任特征的结构。它描述了有效完成特定组织工作所需要的知识、技能和特征的独特结合。对于胜任特征模型的内容或结构，目前存在以下几种理解。

（1）从类别的角度看待胜任特征的内容。斯本塞在提出胜任特征概念之后，界定了胜任特征包含的内容，它们是知识、技能、自我概念、特质和动机。麦克莱兰认为胜任特征除包含上述五个方面外，还包括社会角色，并认为由这六个方面组成的胜任特征模型是分层排列的：知识、技能是显性的，如浮出水面的冰山之一角；社会角色、自我概念、特质、动机是隐性的，是沉入水中的冰山的主体部分。

（2）从不同维度上阐释胜任特征的内容与结构。诺德哈格（Nordhaug）认为，胜任特征包含三个维度：任务具体性、行业具体性和公司具体性。任务具体性（task specificity）是指胜任特征与完成一项具体工作任务的相关程度。低任务具体性意味着胜任特征基本上与任何一项具体工作任务没有特殊关系，而与更大范围内的不同任务关系密切，如分析能力、与他人协作的能力、问题解决能力、沟通技能等。如果一种能力仅适用于某一公司，就是公司具体性（firm specificity）。米切尔和载迪（Mitchell & Zaidi） 认为，公司具体性胜任特征可以导致雇主与雇员因双方利益上的一致而产生一种长期的契约。行业具体性是与某一行业密切相关的胜任特征，某些非公司具体性的特征可以成为行业具体性特征。诺德哈格将任务具体性、公司具体性和行业具体性三个维度结合起来，构成了一个表 15.1 所示的胜任特征分类框架。

表 15.1　诺德哈格的胜任特征分类表

任务具体性	公司具体性		
	低		高
	行业具体性		
	低	高	
低	I 元胜任特征	II 行业通用胜任特征	III 组织内部胜任特征
高	IV 标准技术胜任特征	V 行业技术胜任特征	VI 特殊技术胜任特征

I型：元胜任特征。该类胜任特征是低公司或行业具体性的，可用于完成大量的不同任务，包含广泛的知识、技能和态度。例如，读写能力、学习能力、分析能力、创造能力、感知和操作环境信号与事件的能力、容纳和掌握不确定性的能力、与他人沟通和合作的能力、谈判的能力、应变的能力等。

II型：行业通用胜任特征。该类特征具有低任务具体性和高行业具体性，拥有完成某一行业任务的能力。例如，拥有产业结构及其目前发展状况的知识，具有分析竞争对手战略和运作方面的能力、在行业中同其他公司形成合作和联盟的能力等。

III型：组织内部胜任特征。该类特征具有低任务具体性和高公司具体性的特点，拥有关于组织文化、公司内部沟通渠道、组织中的政治动态和公司的战略及目标等方面的知识。

IV型：标准技术胜任特征。该类特征具有高任务具体性、低行业具体性和低公司具体性特点。它是一个范围较广的具有操作定向的胜任特征。例如，打字和速记技能、普通预

算、计算机编程技能等。

V型：行业技术胜任特征。该类特征具有高任务具体性、高行业具体性、非公司具体性的特点。它在行业内可被跨公司流动使用，并且仅仅可用来完成一个或少量有限的工作任务。例如，建造自动机械、拼接计算机硬件等。

VI型：特殊技术胜任特征。该类特征具有高任务具体性和高公司具体性的特点，仅仅被用来解决一个公司内部的具体任务，与独特技术和日常操作的知识和技能有关。例如，在公司里与使用特殊工具进行精巧制作相关的技能。

（3）从功能的角度看待胜任特征的内容。纵观几十年来关于胜任特征的研究成果，结合近几年工作分析发展的新趋势，有人认为采用上述分类仍显呆板，不能涵盖胜任特征的全部，并且影响了胜任特征运用的灵活性。因此，主张从功能的角度来考察胜任特征的结构，包含两层含义。首先，胜任特征是一个涵盖了KSAO四方面内容的结构。K（knowledge）是知识，指胜任某一工作岗位所必须拥有的知识，在形式上体现为一组包含事实、规则和程序的信息，它是发展岗位能力与形成岗位技能的基础；S（skill）是技能，指能够轻松而准确地完成组织任务的能力，通常指需要借助身体运动、视觉等完成的心理运动类活动；A（ability）是能力，指完成工作所需要的认知能力，知识是这种认知能力的基础；O（other characteristics）是其他，包括个性、动机、态度等。岗位不同，构成其胜任特征模型的上述四要素的比例亦不相同。其次，胜任特征是成功适应工作环境、任务要求的个体KSAO的整合，即个体对工作环境的成功适应（胜任特征）是通过整合能力、恰当的知识、情感以及行为模式等而获得的，是一种整合了的内部结构。从功能上对胜任特征的结构进行界定，使胜任特征具有了情境依存性，即不同的工作环境，可以产生不同的胜任特征，胜任特征是多元的。

三、胜任特征模型的特点

与传统的工作分析相比较，胜任特征模型具有自己的突出特点。

（1）强调“O”的功能作用。在传统的工作分析中，通常以现有的量表（如MMPI、16PF等）测量员工的个性特点，然后与工作绩效进行相关分析，发现个性特点等与工作绩效的相关甚低，由此忽略“O”的作用。胜任特征模型从工作中表现优异者的特征出发，直接收集数据，不受任何现有量表的限制，不但能直接考察知识、能力、技能等的作用，还可以直接考察个性特点、动机、信念等的作用，如工作中的利他行为、团结协作精神等，认为这些行为或精神是影响个体或组织绩效的重要因素，尤其在今天及未来的企业中，当合作变得越来越重要、越来越多的企业摆脱手工操作、依赖个体精神努力时，“O”的作用愈发显得重要。

（2）可以从个体和组织（核心竞争力）两个层面建立胜任特征模型。就目前国内的研究现状来看，多从个人层面建立胜任特征模型，如对电信行业员工胜任特征的研究、对家族企业高层管理者胜任特征的研究、对IT企业项目管理者胜任特征的研究等，都是从某一行业或职位的特点出发，从个体层面整合KSAO，构建模型。随着研究的深入及组织的发

展，建构组织胜任特征模型成为必需。山奇士（Sanchez）等将组织胜任特征定义为以各种方式维持协调、调度资源以实现组织目标的能力。多纳（Donna）将组织胜任特征定义为可能给公司带来竞争优势的知识、技能、技术体系。组织胜任特征通常被称为核心胜任特征（core competencies），它是组织中独特的、具有竞争优势的各种资源的综合，包括组织的战略、知识、技术、技能、价值观、文化等成分，是组织中个人和组织胜任特征的整合，随时间和组织的发展而积累，难以被竞争对手模仿，因此核心胜任特征是组织核心竞争力的重要源泉。多纳采用问卷法对 14 家制药业公司的胜任特征进行研究，发现技术胜任特征、市场胜任特征、协调胜任特征与公司绩效相关。索玛斯（Thomas）用问卷法研究组织的技术胜任特征和网络胜任特征，发现二者对组织改革的成功具有重要影响。

（3）胜任特征模型是未来定向的、规定性的，或者说是针对现在与未来的组织发展策略而建构的特征模型，它通常以组织推崇的价值或经验丰富的管理者的组织发展愿景为出发点，以工作发展的未来趋势对个体或组织 KSAO 的要求为基础。例如，某一组织的胜任特征模型包括下列 10 种胜任力：商业敏感性、沟通能力、团队精神、坚韧不拔、影响他人、关键性思维、管理冲突与变化、结果定向、创新、表现优秀。就目前来讲，这些特征是未来组织发展的必备特征，只有具备这些特征，才能在未来竞争激烈、全球化趋势的商业大潮中求生存、求发展。以这些特征为准绳选拔人才、培训职工，才能使组织占得先机，立于不败之地。而传统的工作分析，由于以现有的工作岗位为基础，得出的结论必然是“过去式”的、总结性的，滞后于组织发展。

（4）操作简单、省时。胜任特征模型只对比在工作中表现优秀者和一般者之间的差异，选择优秀特征构成模型，因此与照顾到方方面面的传统工作分析相比较，更加经济。

四、胜任特征模型的作用

在现代人力资源管理和开发中，运用胜任特征模型可以为组织与企业选拔优秀人才，有针对性地培训人才，灵活设计薪酬和激励员工提供依据。

（1）人才选拔。首先，由于胜任特征是基于工作中表现优秀者的特征，因此以这些特征作为效标选拔人才，可以为组织招募到优秀员工。其次，由于胜任特征是表现优秀者的核心的、潜在的特征，或者说是影响工作绩效的本质特征，因此，以这些特征挑选人才，可以为组织选拔到具有核心竞争力的员工。相对于传统的人员选拔（注重教育背景、知识水平、相关的工作经验）而言，择人标准更加直接，重点突出，避免了仅凭学历水平而选择人才“赝品”的尴尬。第三，胜任特征可以针对未来职位的工作特点做出规定性的描述。当今世界，市场与科学技术的变化日新月异，企业往往需要在短时间内完成转型，企业结构与战略方向的改变会导致新岗位的出现、旧岗位的消失。基于胜任特征的人才选拔可以前瞻性地为组织或企业的不断改组与发展做好后续人才的准备工作。

（2）员工的培训与发展。培训的目的是帮助员工弥补不足，从而达到岗位的要求；培训的原则是投入最小化、收益最大化。基于胜任特征的培训，可以做到量身定做，因此可以帮助员工弥补自身不足，有的放矢地突出培训重点。其次，也是最重要的，最能体现基

于胜任特征模型培训特点的是，胜任特征模型是未来定向的、规定性的，就好比事先已经塑造好了某类人才的模型，根据这种理想模型展开培训，针对性强，便于设定培训的内容与模式，更好地促进员工的发展。例如，布莱克（Black）曾根据全球性领导应该具有的胜任特征模型，提出培训的根本目标：拓宽个体思路，形成一个可以包容全世界的心理图式（mental map）。由此提出培训模式：重构心理图式，即用某些新观念影响旧有的心理图式，以使它迅速解体并形成新的心理图式。

（3）薪酬设计。基于胜任特征模型的薪酬设计奖惩分明，便于量化，与传统的薪酬设计相比较，更有利于促使员工产生组织所需要的行为，激发员工的工作热情，形成良性循环。

五、建立胜任特征模型的方法

建立胜任特征模型的方法有许多种，目前最常用的是行为事件访谈法（behavioral event interview，BEI）。该方法是由美国心理学家麦克莱兰提出来的，他将工作分析中的关键事件法和主题统觉测验结合起来，采用开放式的行为回顾式探察技术，让被访谈者找出他们在工作中最成功和最不成功的三件事情，进行详细描述，然后分析访谈内容，通过对比担任某一任务角色的卓越成就者和表现平平者体现出的特征差异，最终确定该任务角色的胜任特征模型。行为事件访谈法是目前得到公认的、建立胜任特征模型的最有效的方法，另外的方法还有专家小组法、问卷调查法、观察法等。

第二节 领导者胜任特征模型

一、领导者胜任特征模型的含义

领导者胜任特征模型是运用上述胜任特征模型的理论及建模方法研究领导有效性或领导力的一种方法。在第十三章中已经详细介绍了许多领导有效性的理论。与那些理论不同的是，领导者胜任特征模型研究另辟蹊径，从具体领导岗位出发，以领导者自身特征为主要研究对象，试图甄别有效领导者的主要特征。它与过去的领导有效性理论具有以下几点区别：

（1）两者的概括水平不同。过去的所有领导有效性理论都试图建立一种普适性的理论。例如，变革型领导认为变革型领导应该能够“预见未来趋势，激发追随者理解并包容一种新的可能性的愿景，使追随者成长为领导者或更好的领导者，把组织或团体建设为一个受到挑战与回报的学习者的集体”。变革型领导理论是当今社会经济和组织发展对领导力诉求的一种理想表达，普适于政治、经济、文化领域的所有领导者。而领导者胜任特征模

型只研究某一具体领导岗位所需要的领导特征，具有具体性的特点。例如，姚翔、王垒等人对项目管理者胜任特征的研究表明，项目管理者应该具有计划性强、全局观念、沟通能力、协调能力、责任心强、勇于承担责任等 23 个胜任特征；仲理峰、时勘对家族企业高层管理者胜任特征的研究表明，我国家族企业高层管理者应该具有自信、主动性、信息寻求、组织意识、影响他人、权威导向、仁慈关怀等 11 项特征。

（2）两者研究的范围不同。过去的领导有效性理论涵盖领导有效性的方方面面，只要对领导有效性有所贡献的方面都可以包含在内，包括环境、任务、个性品质等。例如，领导者的权变理论认为领导有效性并不取决于领导者自身的行为和个性，而取决于领导者、被领导者和环境三者之间的关系，领导有效性是三者的函数（参见第十三章）。而领导者胜任特征模型只关心领导者自身拥有的个性及行为特征，而不涉及其他方面，具有专一性的特点。

（3）两者的出发点或针对性不同。过去的领导有效性理论主要是针对现在的，是为了说明和解释现有领导者的领导有效性问题，弄清楚它的影响因素，为现有的领导者及其上级出谋划策。而领导胜任特征模型则主要是针对未来的，是为聘用和选拔未来领导者服务的，具有前瞻性的特点。

尽管领导者胜任特征模型与过去的领导有效性理论具有显著的不同，但它们之间也存在着千丝万缕的联系。领导者胜任特征模型常常以过去的领导有效性理论作为自己建模的理论基础和参照。例如，奥德基（Alldredge） 等为 3M 公司建立的全球性领导者胜任特征模型由 3 类 12 个胜任特征组成：第一类是根本的胜任特征，是新职工被雇佣时就应该具备的特征，包括伦理道德和真诚、智力技能、成熟和判断；第二类是必需的胜任特征，这些特征可以为提升到更高和更复杂职位做准备，包括消费者定向、培养和激励他人、商业健康和结果；第三类是远景性（visionary）胜任特征，包括全球性观念、远景和战略、培育创新、建立联盟、组织灵活。而根据伯恩斯（Burns） 和巴斯（Bass） 对变革型领导的界定，变革型领导应该具有下列行为特征：个性魅力、激发他人动机、启迪他人智能、个人关怀等。其中激发他人动机是指创设愿景，描述组织的美好未来，以使追随者获得工作的意义和迎接工作中的挑战；启迪他人智能是指能够质疑问题假设、重塑问题结构以及产生崭新观点和解决问题方法的能力。对比奥德基为 3M 公司建立的全球性领导者胜任特征模型和伯恩斯等人提出的变革型领导理论，不难看出二者之间的联系。

二、领导者胜任特征模型的现有研究成果

领导者胜任特征模型和领导有效性理论的差异决定了领导者胜任特征模型存在的必要性，它的具体性、专一性和前瞻性等特点为领导选拔和培养提供了切实可行的依据和工具。因此，领导者胜任特征模型的研究范式一经产生，便得到迅速发展，目前国内外已有大量研究成果。

1970 年，McBer 公司和美国管理协会对 1800 名管理者进行了长达 5 年的跟踪研究，结果识别出优秀管理者的五个重要胜任特征：专业知识、心智成熟、企业家成熟度、人际

间成熟度、在职成熟度。1982年，理查德（Richard Boyatzis）对12个行业41个管理职位的2000多名管理人员进行研究，建立了管理人员胜任特征的通用模型，该模型包括6个特征群（目标和行动管理、领导、人力资源管理、指导下属、关注他人、知识）和19个子胜任特征（效率定向、主动性、关注影响力、判断性地使用概念、自信、概念化、口才、逻辑思维、使用社会权力、积极的观点、管理团队、准确的自我评价、发展他人、使用单向的权力、自发性、自控、自觉的客观性、精力和适应性、关注亲密的关系）。Hay Group公司利用自己遍布全球的分公司力量，建立了丰富的胜任特征模型库，认为优秀领导者的胜任特征包括四个因素：信息搜寻、概念思维、战略定向和客户服务定向，并开发了相应的领导胜任特征问卷。阿尔法（Alpha Assoc） 和沃特鲁（Waterloo）总结出管理人员具有的5项基本胜任特征：概念技能与独创性、领导、人际技能、行政管理和技术。范特南（Virtanen）对公共经理进行了公共事业管理胜任特征的研究，提出一个包含5大胜任特征的模型：任务胜任特征类、实体政策领域的职业胜任特征、行政管理方面的职业胜任特征、政治胜任特征和伦理道德类胜任特征。美国明尼苏达大学的研究人员经过多年的研究和实践，提出了领导者应具有的20多种胜任特征，如获取信息的技能、分析思考的技能、概念思考的技能、策略思考的技能、人际理解和判断的技能、帮助服务定向的技能、对他人的影响技能、对组织的知觉技能、建立和管理人际关系的技能、发展下属的技能、指挥技能、小组工作和协作技能、小组领导技能等。多维奇和赫伯特（Dulewich & Herbert,1999）对大不列颠和爱尔兰一些公司的总经理进行了7年的跟踪实证研究，得出12大类胜任特征因素：战略眼光、分析与判断、计划与组织、管理员工、说服、坚持性与果断、人际敏感性、口头沟通、毅力和适应能力、精力与主动性、成就动机和商业意识。

我国对胜任特征模型的研究虽然刚刚起步，但其应用前景已得到政府、企业组织以及一些人力资源管理公司的认可，如北京市西城区宣布2006年将运用胜任特征模型作为辅助手段选拔正处级干部。国内一些研究者也正致力于各级各类胜任特征模型的研究与建立工作，并产生了一些成果。如前面提到的姚翔、仲理峰等人对项目经理及家族企业高层管理者胜任特征模型的研究，时勘等人对企业高层管理者胜任特征的研究，该研究得出企业高层管理者应该具有20项胜任特征：成就欲、关注质量与秩序、主动性、信息寻求、人际洞察力、客户服务意识、影响力、权限意识、公关、发展他人、指挥、团队协作、团队领导、分析性思维、概念性思维、技术专长、自控、自信、灵活性、组织承诺等。

目前胜任特征模型的研究方兴未艾，期望将来的研究能够更加细化、科学化，为各级各类领导者选拔和培训提供可靠基础。

第三节　领导干部选拔

领导干部选拔是一项严肃而复杂的工作。传统的领导干部选拔有一套严格的组织程序，通常采取两种方式：一种是自上而下的组织任命方式，即由上级组织经过培养、考察、民主评议、个别谈话、研究讨论等步骤，最终决定领导干部的人选；另一种是自下而上的选

举方式，即完全依靠群众投票选出领导干部。也有些组织采用介于这二者之间的选拔方式。目前，我国党政机关、事业单位、国有企业的领导干部选拔基本上沿袭这种程序，但也出现了一些新的做法。例如，通过公开招考方式（如公务员考试、局处级干部招考等）选拔领导干部，这种方式通常也会结合一定的组织程序（如考试资格审查、政审等），有时候还会采用一些与心理学相关的测试手段（如面试、无领导小组讨论、结构化访谈等）。而在非公有制企业或组织（如外资企业、私营企业、私立学校、私营机构）中，领导干部的选拔则采用不同的程序和方式，总体上看，其特点是目标更加明确化（如岗位职责具体明确）、程序更加公开化（如向社会公开招聘）、形式更加多样化（如依靠猎头公司、专业人才评价机构、高校专家等；采用书面考试、面试、各种形式的心理测验等）。下面着重对近些年来涌现出来的、在企业领导干部选拔中经常采用的几种心理学方法加以介绍。

一、360 度反馈评估

360 度反馈评估（360-degree feedback assessment），又称多评价者评估（multi-rater assessment）、多源反馈（multi-source feedback，简称 MSF）或全方位评价（full circle appraisal）。它是一种全方位整合信息的评价形式，包括上级评价、下属评价、同事和自己评价，有时还包括由顾客和与受评者有业务关系的其他人的评价。与传统的自上而下评价方式相比，其本质特点是信息来源的多样性，从而可以保证反馈的准确性、客观性和全面性。

360 度反馈评估方法有两个最基本的前提假设：（1）对个体从多角度进行观察将得出更有效和可靠的结论。首先，个体行为是立体的、多方位、多层次的，传统的自上而下评估方式只能提供被评估者的单方面信息。如果把一个人比作一座山的话，“横看成岭侧成峰，远近高低各不同”，不同的人观察的角度不同，看到的景象就会不同。即便是上级主管，也只能看到被评估者的某一侧面。而下属或同事与被评估者之间有着不同的交往和关系，他们可以提供一些从上级主管那里难以获得的关于被评估者的信息；同样，来自顾客或其他相关人士的评价也会很不相同。研究表明，同事能够提供最可信的评价，其次是下属，最后才是上级。其次，多源反馈可以减少误差。研究表明，如果所有参与评价者事先被告知他们的评估结果会与其他人的评估结果相比较，他们就会在评估过程中表现得更加客观。（2）行为的改变贯穿于被评估者自我意识增强的过程之中，如果自我意识改变了，其行为也将发生改变。360 度反馈评估通过反馈向被评估者表明“什么是需要学习和改进的”，使他们可以据此改变自己的行为。理查德等人（Richard ,James & Nicholas,1996）的研究发现，在收到下属评价的反馈结果后，那些最初绩效较低的管理者的绩效得到了明显的改善，并在随后的 2 年内持续保持改进。史密塞（Smither）等的研究也表明，最初收到负面反馈（也就是最低的评分）的管理者，其行为的改善最大。他们认为，这种现象可以用“自我一致性理论”加以解释。当管理者发现他人对自己的评分低于自我评分时，就会考虑自己的实际行为是否与其自我意象相一致，并为了维护其自我印象而试图改善绩效，以减小自我认知与他人印象之间的差距，恢复认知平衡。研究者指出，自评与他评一致性

和评价者之间一致性（如下属之间、上级与同事之间）的交互作用将影响被评估者对反馈的接受程度和态度，被评估者越是愿意接受反馈，其绩效越可能得到提高。

目前用于360 度反馈评估的主要工具是问卷调查，因此问卷设计的合理性和科学性直接影响 360 度反馈评估的效果。为了保证评估的可靠性，在设计问卷时应注意以下几个方面的问题：（1）以工作分析的结果、组织倡导的价值观、组织期待的工作行为等为标准设定项目。例如，通用公司设计问卷的标准包括眼界、对顾客/服务质量的关注程度、完整性、责任心/投入与承诺、沟通/影响力、组织协调/授权、知识/能力/智力、主动性/快速性等。（2）根据不同的评估目的设定不同的项目。对于领导者的选拔而言，偏重于对个体能力的诊断与评估，在项目设定上应侧重于考察个体的潜在品质与能力。许多研究者建议此种情况下应基于胜任特征模型构建 360 度反馈评估问卷。胜任特征本身虽然不能直接作为评价指标，但它与各种管理行为指标有直接联系，借助胜任特征模型能产生相应的行为评价指标。就目前情况来看，国内外研究和实践所使用的问卷结构千差万别，尚缺乏系统的评价结构。（3）问卷的长短对评价效果具有影响作用。问卷的设计不能过长，否则会加大评估者的考核误差，尤其当评估者考核多个对象时，表现更为明显。但问卷也不能过短，否则不能包含所有相关行为项目，会影响评估的质量。问卷的长度一般以评定者能在 10～15 分钟内完成为标准。（4）必须坚持反馈的匿名原则。伦丹（London）等人在研究中发现，如果反馈不是匿名的，约有 24%的人表示将要更改其对上级或同事的评价。多数研究者都建议用第三方来分析数据和准备反馈报告，如外部咨询公司或专业的管理顾问公司。

360 度反馈评估也存在不足之处：（1）由于评价参照标准的不确定性，往往使得评价只关注一般特质，而不是特定工作行为；（2）评价是以个体记忆为基础的，不能真实反映被评估者过去的工作行为；（3）评价者不能观察到被评估者的全部工作行为，易以偏盖全；（4）在实施 360 度反馈过程中，如果培训和运作不当，可能会在组织内部造成紧张气氛，影响组织成员的工作士气。

二、评价中心技术

评价中心（Assessment Center）是近年来被广泛使用的一种人员选拔方法，它把已考虑雇佣或提升的候选人放置到模拟的工作环境中，以观察和评价他们在压力情境下的行为表现。评价中心是二战后迅速发展起来的一种人员测评技术，它应用现代心理学、管理学、计算机科学等相关学科的研究成果，通过无领导小组讨论、公文筐、角色扮演等情境模拟技术，对人的知识、能力、个性、动机等进行测量，从而实现对人的个性、动机和能力等较为准确的把握，做到人—职匹配，确保人员达到最佳工作绩效。

评价中心起源于德国 1929 年建立的一套用于挑选军官的多项评价程序，在这套评价程序中评价人员可以自由使用多种方法对行为样本做出评价。二战期间，英国及美国军队在模仿德国评价活动的基础上对其进行改造和发展，综合应用无领导小组讨论、团队任务、

深度面谈等方法和技术，把被测评对象置于更现实的环境中进行评价。1945 年，英国文职人员委员会采取八种评选方法选拔文职人员，包括一组语言和非语言测验、面谈、资格考试成绩、个人和小组的情境模拟练习等。该委员会最先把评价中心应用于非军事目的，并把情境模拟测验的本质变成反映领导和管理才能的测评方式，他们的工作为评价中心最终应用于工商业领域创造了条件。今天广泛使用的评价中心的基本模式来源于美国电报电话公司（AT&T），当时布雷在 AT&T 公司受雇负责一个管理发展研究的项目，该项目的目的是获得那些正在成为或试图成为公司中高级经理人才的特点和成长状况的资料。他们应用包括面谈、公文处理练习、商业游戏、无领导小组讨论、投射测验（句子完成测验）、自我描述等方法获取上述资料。之后，美国许多著名大公司如通用电气公司（GE）、福特汽车公司（Ford）等也都采用此种技术，并建立相应的评价机构来评价管理人才。1975 年 5 月，在加拿大魁北克举行第三届评价中心国际大会，制定并通过了《关于评价中心的实施标准和道德准则》，评价中心技术走向规范化。我国对评价中心技术进行系统研究和实践应用是 20 世纪 80 年代后期的事情，许多跨国公司的管理实践和学术界的介绍推动了评价中心在我国的发展。20 世纪 90 年代后期，随着市场经济的发展、企业对选才工作的重视以及众多管理咨询公司的成立，评价中心技术逐步被应用于我国企业的人力资源管理工作中。

1．文件筐技术（in-basket technique）

文件筐技术是评价中心常用的核心技术，要求求职者处理某一特定职位（如职业经理人）公文包中的备忘录或信件等。公文包的材料包含典型的疑难问题、问询或详细指令等，求职者必须在规定时间内处理好这些材料。评估者通过分析其对材料的处理情况，评价其潜在能力。传统的公文筐技术是以测评对象扮演管理者批阅公文的形式进行的，目前越来越多的公文筐测验改用邮件形式进行，即通过让测评者扮演背景材料中的角色，并作为这个角色处理收到的邮件，测评师根据测评对象如何回复这些邮件做出评价。

2．无领导小组讨论

这也是常用的评价中心技术之一。将被评价人（通常 4~8 人）组成一个小组，让他们在给定时间内讨论某一实际的业务问题，在小组中不指定领导，也不安排特定角色，整个讨论过程完全由小组成员自行安排。例如，在一次 AT&T 组织的无领导小组讨论训练中，小组成员为 6 人，告诉他们假设自己是企业的经理并且要在限定时间内为企业提高利润。给予他们关于公司和市场的信息，但不指定领导者，也没有建立怎样完成目标的规则，请他们讨论。假定某位小组成员在讨论中行使了领导者的权力，这个人的能力可以被评估，组中其他成员在执行领导分配任务中的合作情况也可以被评估。

3．角色扮演

要求被评价者每人扮演一个角色，模拟实际工作中的一系列活动，借以考察他们多方面的能力，如组织协调能力、合作能力、实际工作能力、处理突发事件的能力等。角色扮演提供了一对一的人际情境，较之小组讨论更能够体现测评对象的一些人际技巧，如人际理解、行为塑造和说服能力等。

除此之外，评价中心技术还包括管理游戏、案例分析和演讲等。

上述评价中心的不同方法在测量能力方面各有所长。在长期的评价实践中，人们积累了一些经验，发现了一些最佳匹配，如表 15.2 所示。

表 15.2 不同管理技能的最佳测评方法

评价指标	最佳的测评方法
经营管理技巧	文件筐作业
人际关系技巧	无领导小组讨论、管理游戏
智力状况	纸笔测验
工作的恒心	文件筐作业、无领导小组讨论、管理游戏
工作动机	投射测验、面试、情景判断测验

评价中心具有较高的预测效度，表 15.3 列出了 1970—1987 年间一些研究者对评价中心预测效度的研究结果。从中可以看出，各项评价中心技术的总评结果具有较高的效标关联效度。

表 15.3 总评结果（overall assessment rating，简称 OAR）预测效度的研究

研 究 者	时 间	结 论
Byharn	1970	OAR 能够找出那些步步高升的管理者。从测评结果来看，成功管理者的得分高于不成功的管理者。OAR 与业绩的相关系数范围从 0.27 到 0.64
Cohen 等	1974	OAR 与业绩的相关系数为 0.33，与潜能的相关系数为 0.63，与职业晋升的相关系数为 0.40
Thornton&Byham	1982	OAR 能够预测职业晋升情况、业绩水平等
Hunter&Hunter	1984	统计分析的结果表明 OAR 与工作业绩的相关系数为 0.43
Schmitt 等	1984	统计分析的结果表明 OAR 与一系列才能指标的相关系数为 0.41
Gaugler 等	1987	统计分析的结果表明 OAR 与晋升、工作业绩评价等的相关系数为 0.37

虽然评价中心具有较好的效标关联效度，预测准确性很高，但其结构效度却不太理想。评价中心的结构效度一般有两个指标：（1）汇聚效度（convergent validity）：指的是同一评价维度的评分在不同测评方法之间的相关性。例如，无领导小组讨论、公文筐、演讲等不同的测评方法都包含了对“工作组织性”这一维度的评价，它们之间的相关系数越高，说明汇聚效度越好；（2）区分效度（discriminant validity）：指的是同一测评方法在不同测评维度上评分的区分性。例如，无领导小组讨论一般包含对敏感性、工作组织性、问题解决能力、领导能力等不同维度的评价，这些维度得分之间的相关越低，说明区分效度越好。萨克特等（Sackett & Tuzinski）的研究发现，在典型的评价中心中，同一维度在不同测评方法间评分的平均相关为 0.25，而相同测评方法中不同维度评分的平均相关却为 0.58。在同一测评活动中，不同维度评分间的相关要明显高于不同测评活动中相同维度评分间的相

关，也就是说，测评活动的影响效果仍然要明显大于测评维度的影响效果。

近些年的研究对评价中心缺乏构思效度给出了几种解释：（1）测验方法偏差。把评价中心结构效度不理想的原因归结为测量方法上的偏差。一种观点认为测验的汇聚效度和区分效度之所以较差，是因为考官在测验中的认知加工负荷太高，难以做出精确判断，因此建议减少测评维度、降低维度构思间的重叠、采用行为列表式计分方法等。（2）测评情境导向。朗斯（Lance）等认为，在过去的研究中普遍发现了测评活动的显著影响，恰好反映了被评价人员的绩效表现是以测评情境为导向的，而非方法偏差。持这种观点的研究者认为，评价中心结构效度的缺陷不应归因于程序和设计上的问题，影响评价中心测评结果的是测评情境。有研究者认为，评价中心本身包含的测评活动，不能真正探测到所要测量的结构，相反，测量到的更多是一些迎合测评活动本身的能力。例如，善于表演的能力、情境适应性等等。（3）内隐绩效视角。一些研究者认为，评价中心缺乏构思效度可能是因为测量到的构思并不是设计者既定的构思，因此其根源在于对评价中心构思域中潜在结构的界定偏差。这种观点随着内隐领导理论的发展而逐步得到更多人的接受。内隐领导理论的相关研究指出，考官（评价者）对领导模型和绩效模型的内隐认知驱动了其对考生（被评价者）的评价；评价中心测量到的可能是考官的内隐领导概念。

三、结构化面试与非结构化面试

结构化面试和非结构化面试是面试的两种不同形式。面试是一种经过精心设计，在特定场景下，主试与应试者双方以面对面地观察、交谈等双向沟通方式，由表及里测评应试者有关素质的一种方式。它具有对象单一、内容灵活、直接互动、信息丰富等特点，是近几年来人员选拔工作中经常使用的手段与工具。

（一）非结构化面试

非结构化面试是指没有事先准备的面试提纲和固定模式的一种面试方式。安排的形式和提出的问题由测评人员的判断决定，问题多是开放性的，在双方自由交谈的过程中，主试观察应试者的组织能力、知识面、谈吐和风度等。由于非结构化面试更多地依赖于主试，不同的主试可能会对应试者的背景、经验或态度的不同方面感兴趣，反映的更多是自己的偏向或偏见，而不是应试者的客观资格，因此各测评人员之间的评分常常缺乏一致性，从而影响了其信度和效度。早在 1929 年，贺林沃斯（Hollingworth）就曾对测评者评分一致性问题进行过研究。他分别调查了 12 位测评人员，让他们对 57 名申请人的销售能力进行打分。结果发现评分者之间严重缺乏一致性，被某位测评者列为第一的申请人常常被另一位测评者排在了最后。图 15.1 显示了 12 位测评人员对其中一位申请人的判断结果，可以看出对这个申请人的等级评判从最低值 1 到最高值 55 均有，评判的不一致性可见一斑。

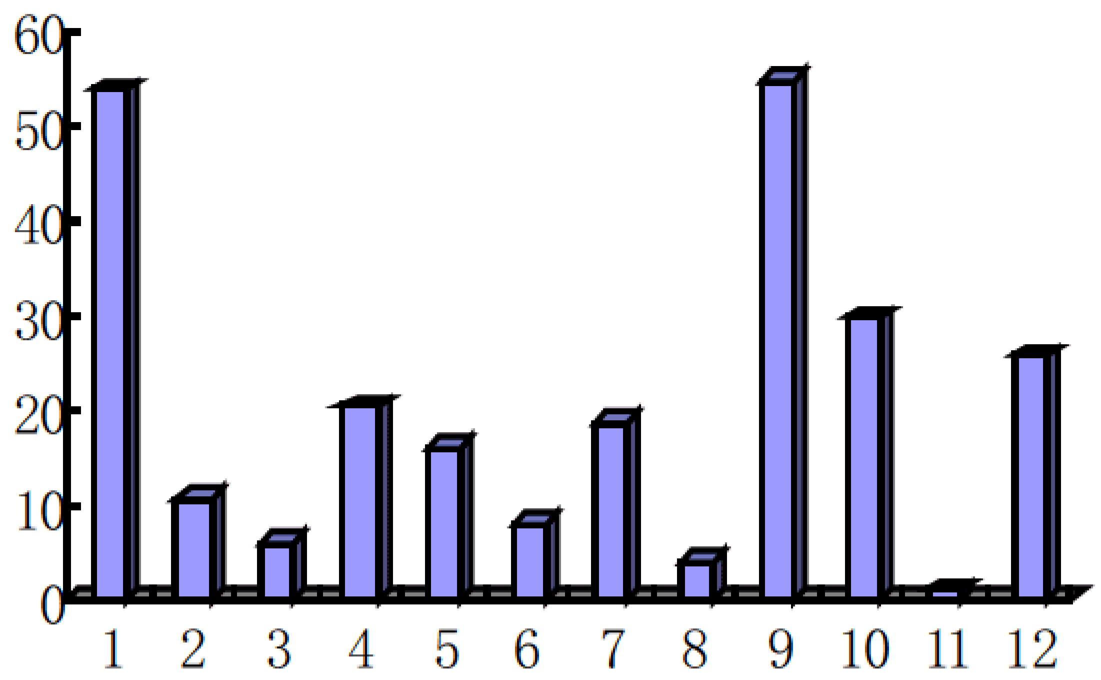


图 15.1　12 位测评人员对一名申请人的评定

（二）结构化面试

结构化面试，也称标准化面试，是根据事先制定的评价指标，预先设计好问题、评价方法和评价程序，并严格按照计划进行的面试。例如，宝洁公司对应聘者进行面试时通常询问的问题如下：（1）请你举一个具体的例子，说明你是如何设定一个目标然后达到它的；（2）请举例说明你在一项团队活动中是如何采取主动，并且起到领导作用，最终获得你所希望的结果的；（3）请你描述一种情形，在这种情形中你必须去寻找相关的信息，发现关键的问题并且自己决定依照一些步骤来获得期望的结果；（4）请你举一个例子说明你是怎样通过事实来履行你对他人的承诺的；（5）请你举一个例子，说明在完成一项重要任务时，你是怎样和他人进行有效合作的；（6）请你举一个例子，说明你的一个有创意的建议曾经对一项计划的成功起到了重要作用；（7）请你举一个具体的例子，说明你是怎样对你所处的环境进行评估，并且能将注意力集中于最重要的事情上，以便获得你所期望的结果的；（8）请你举一个具体的例子，说明你是怎样学习一门技术并且怎样将它用于实际工作中的。测评人员按照上述问题及问题的顺序进行面谈，即是一次结构化面试。通常情况下，上述问题会被打印成一份表格，面试时考官只需按照表格进行提问，并将申请者的回答如实记录在同一张表格中。这样会使面试更加规范，减少考官在面试过程中的随意性、主观性，避免不同应聘者遇到不同的程序和问题。

由于其程序的标准化，与非结构化面试相比，结构化面试具有更高的预测效度。斯米特和洪特（Schmidt & Hunter）的元分析结果表明，结构化面试效度系数与认知能力测验的效度相同，都达到了 0.51，而无结构化面试的效度系数为 0.38。结构化面试也具有较高的信度，研究发现，当通过电话途径进行结构化面试时，同样能够像面对面那样对应聘者将来的工作绩效做出有效的预测。结构化面试较高的信、效度使其成为领导干部选拔与评价方法体系的重要组成部分，是测定被试领导能力和个性特征与相关职位匹配程度的有效方法。近年来，领导干部结构化面试在制度化建设、职位针对性以及考务规范化等方面有较大的发展，实际效果也比较好，得到了广泛的应用。

本章小结：

1．胜任特征是能将某一工作岗位上表现优秀者和表现一般者区分开来的个体潜在特征。胜任特征模型是指担任某一特定任务角色所需要具备的胜任特征的总和。可以从胜任特征的类别、维度、功能等角度分析胜任特征的内容和结构。与传统的工作分析相比较，胜任特征模型具有自己的突出特点。胜任特征模型可以为组织选拔优秀人才、有针对性地培训人才等提供依据。建立胜任特征模型最常用的方法是行为事件访谈法。

2．领导者胜任特征模型是运用胜任特征模型的理论及建模方法研究领导有效性或领导力的一种方法。它与过去的领导有效性理论有区别也有联系。目前国内外关于领导者胜任特征模型的研究方兴未艾。

3．近年来在企业领导干部选拔中经常采用的几种心理学方法有 360 度反馈评估、评价中心技术、结构化面试和非结构化面试等。

4．360 度反馈评估是一种全方位整合信息的评价形式。它具有信息来源多样性、准确性、客观性和全面性等特点。用于 360 度反馈评估的主要工具是问卷调查，问卷设计的合理性和科学性直接影响 360 度反馈评估的效果，可以采用一些方法保证问卷设计的可靠性。360 度反馈评估也存在一些不足之处。

5．评价中心技术是把已考虑雇佣或提升的候选人放置到模拟的工作环境中，以观察和评价他们的评价方法。文件筐技术、无领导小组讨论、角色扮演、管理游戏、案例分析和演讲等都是评价中心可以采用的具体技术。研究表明评价中心具有较高的预测效度，但其结构效度却不太理想。

6．结构化面试和非结构化面试是面试的两种不同形式。非结构化面试是指没有事先准备的面试提纲和固定模式的一种面试方式。其问题是各测评人员之间的评分常常缺乏一致性，从而影响了其信度和效度。结构化面试是根据事先制定的评价指标，预先设计好问题、评价方法和评价程序，并严格按照计划进行的面试。它减少了考官在面试过程中的随意性、主观性，与非结构化面试相比，具有更高的预测效度，得到了广泛的应用。

本章思考题：

1．简述胜任特征和胜任特征模型的概念。

2．试分析胜任特征模型的结构和特点。

3．试分析领导胜任特征模型与领导有效性理论的区别。

4．如何保证 360 度反馈评估法的可靠性?请设计一份 360 度反馈评估的问卷。

5．试分析评价中心技术的信、效度。如何提高其信、效度？

6．设想一家企业邀请你帮助他们用评价中心技术选聘人员，请你制定一份详细的实施方案。

7．结合实际设计一份结构化面试的样题，并给出详细的评分标准。



第六篇　组织管理

第十六章　组织的基本概念与理论

组织作为不同于个体的行为主体，有其自身的运作机制和行为规律。组织及其运作机制和行为规律是管理心理学研究的重要内容之一。本章拟首先对组织的概念、组织结构及组织中的团队进行分析。

第一节　组织的概念及功能

一、组织的基本概念

组织一般可作两种解释。其一为动词，是指为了实现特定目标或完成特定任务，按照一定顺序或程序开展的一系列活动；其二为名词，是指为了实现特定目标、按照一定形式建立起来的结构。其中，广义的组织可理解为由诸多要素按照一定方式相互联系形成的系统，包括各种各样的物理结构、生物组织、机体组织、动物的群体组织及由人构成的各种组织等。管理心理学研究的组织是狭义的组织，是指人们为实现特定目标相互依赖、彼此协作而形成的机构或团体，其要点有四：首先，组织就是聚集在一起的一些人；其次，这些人有共同的目标；第三，这些人相互依赖，需要按照特定的规则或规范彼此协作；第四，这些人需要特定的空间和条件开展活动。其中目标或目的的共同性或一致性是组织的本质特征。

二、组织的功能

（一）组织的社会功能

组织源于人类的生产与社会实践。在长期的实践活动中，为了实现一定的目标，人们产生了与他人发展协作关系，创造群体合力，并不断优化这种关系以提高群体效能的需要，而组织正是人们为满足或实现这种需要而努力的结果。在现代生活中，组织已成为社会的基本单元，其影响已渗入到社会领域的方方面面，它凝聚了个体的智慧，形成了推动社会进步与发展的一股股力量。

（二）组织的管理功能

一方面，组织负有协助政府对人们进行管理的职能，各级各类组织本身就是政府管理公民的重要途径或工具。组织是联系个人和社会的桥梁和纽带，正是通过一个个组织，人

们的生活才变得井然有条，社会才得以安定。另一方面，组织自身的目标也要靠组织来实现。组织既是各种管理活动展开的舞台或背景，又是实施和推行管理措施的主体或依托。

（三）组织的个人功能

组织不仅可以为个人带来收入和经济保障，更重要的是可以为个体带来归属感、价值感。组织为个人提供了施展自己抱负，发挥自己才华，实现自身价值的舞台，可以满足个人的自我实现需要。

三、研究组织的意义

研究组织及其运作机制和行为规律有两个层面的意义。对于组织管理者来说，有助于更加科学有效地设计、建立、规范和管理组织，更好地发挥组织的功能。对于生活和工作于组织中的个体来说，可以帮助他们更好地理解和预测组织中发生的事情，有效地参与和影响组织事件，质疑或重新建构自己的组织概念和理论，以便在组织情境中更加有效地工作。此外，从学术意义上看，研究组织还有助于积累关于组织的系统知识，形成完备的组织理论。

第二节 组织结构

一、组织结构及其构成要素

（一）组织结构的概念

组织结构（organizational structure）是指组织内部的工作分配以及指导组织活动的协调、沟通、工作流程和正式权力的模式。组织结构在一定程度上反映了组织内部的文化与各种权力关系，是组织设计的重要依据。形成组织结构应具备两个必要条件：一是工作分配；二是工作协调。工作分配导致社会分工，使任务专门化，而工作协调则有利于维持组织内部工作节奏的和谐与统一。常见的工作协调机制主要包括非正式沟通、正式层级管理与标准化。非正式沟通通常比较适用于小规模组织的管理，它对于组织成员分享信息、形成共同的心理模型、维持工作活动的正常化具有重要作用，此外由于非正式沟通手段的多样性与灵活性，因此它还是应对非常规情境的重要协调机制。正式层级管理是指将合法权力赋予个体，由权力执行者配置资源并指导工作进程。由于正式层级管理主要通过结构性较强的层层监督来协调工作，因此比较适合较大规模组织的管理，但是正式层级管理也存在一些潜在的问题。例如，一个管理者可有效进行监督的员工人数往往是有限的，而且有时命令链的层层传达不如员工间直接交流来得及时与准确。此外，这种等级森严、以权制人的

管理方式容易受到那些崇尚自由的员工的质疑和挑战。标准化是指建立行为或产出的常规模式，其方法主要包括制定标准化工作流程、确立标准化任务目标以及执行标准化操作技能等。

（二）组织结构的构成要素

基于组织中的工作分配与工作协调机制，组织结构主要包括以下四个基本要素：

1．控制幅度（span of control）

控制幅度是指在组织层级管理中直接向上级报告的下属的人数，它决定着组织内部层级的设置与管理人员的配备。根据控制幅度的规模与分布，可以将组织结构区分为纵高型（tall structure）与扁平型（flat structure）两类（参见图 16.1）。纵高型组织内部分工明确、结构严密、等级森严、便于监管。但随着经济全球化的发展，纵高型组织结构的弊端已日益突显。而扁平型组织结构由于减少了组织层级，增大了控制幅度，因此能够有效节省组织的管理成本，提高上下层之间信息传递的效率，并有助于实现工作内容的丰富化，因此扁平型组织结构已为越来越多的组织管理者所采纳。

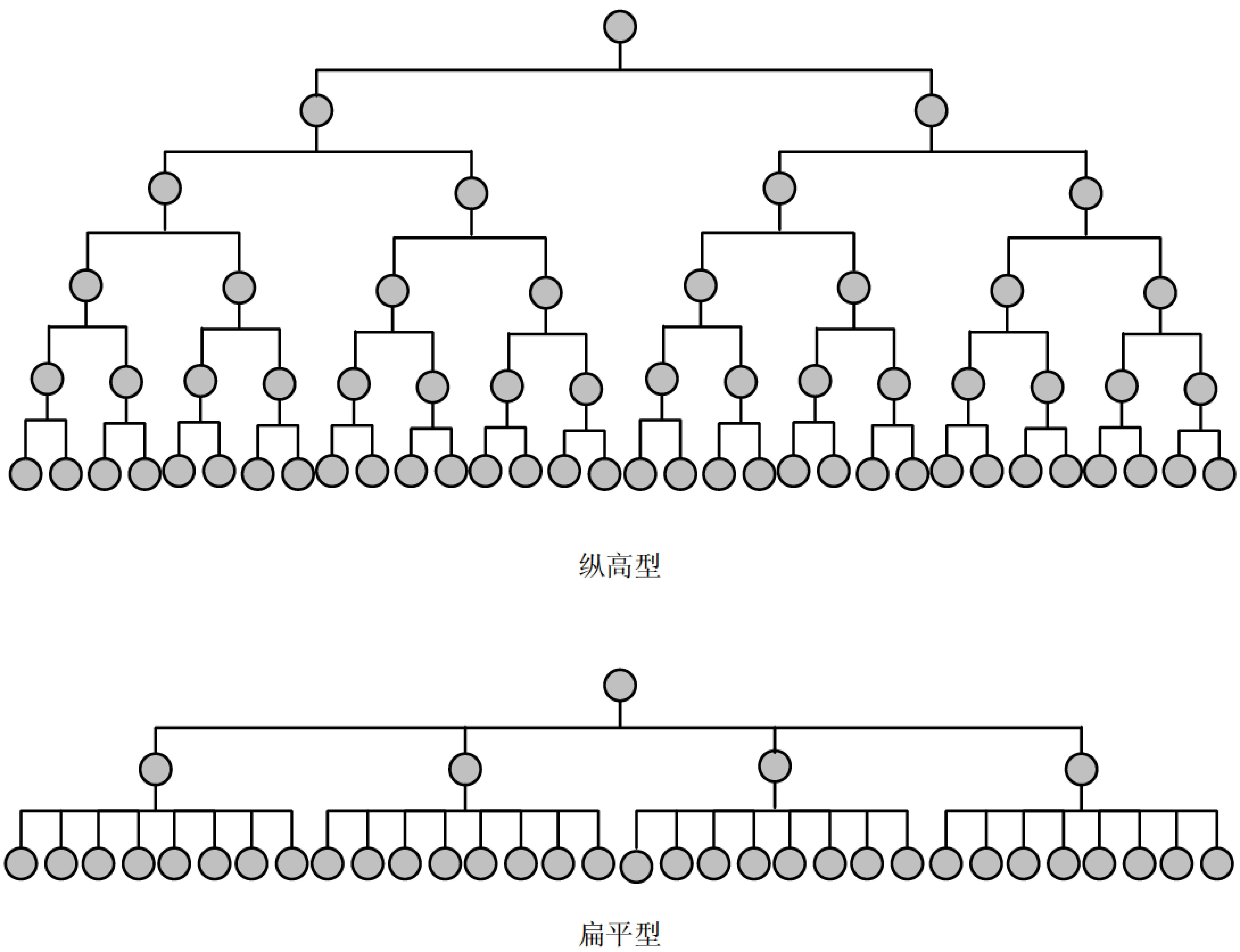


图 16.1　纵高型与扁平型组织结构（引自 Steven L. McShane & Mary Ann Von Glinow,2000）

2．集权化（centralization）

集权化是指在组织层级管理中由少数几位高层管理者掌控正式决策权。在组织建立之初，创建者通常会采取集权化管理，以确保组织的正常运作与目标的实现。在组织发展的关键时期，施行集权化管理也有助于增强领导者对组织事务的控制力。但是集权化所带来的权力过分集中问题也不容忽视；此外，随着组织的发展与壮大，工作活动的细化与业务范围的拓展往往会使领导者感到力不从心，因此与集权化相比，在应对组织内部发展与外部环境的复杂性方面，分权化通常更具有优势。所谓分权化（decentralization）是指将组织决策权散布于不同的层级。分权化管理若运用得当，则不仅能够弱化权力过分集中所造成的不利影响，而且能够调动下属的工作热情，减轻高层管理者的决策负担，促进权责结合。

3．正规化（formalization）

正规化是指通过各种规章制度、工作程序、正式培训等方式实现组织内部操作的标准化。由于采用了标准化的工作流程、任务目标及操作技能，因此正规化比较适用于常规工作的管理。但是，正规化管理也存在一些潜在问题。例如，严格规定的规章程序容易削弱组织活动的弹性，不利于应对瞬息万变的发展形势。此外，一些繁复的制度有时反而会影响组织效率。

4．部门化（departmentalization）

部门化具体规定了如何将员工及其工作活动归为一个组群。通常可以根据职能、产品或地理位置等因素对组织进行部门化管理。部门化管理是协调组织活动的重要机制，它有利于在组织内部确立监督指导体制，建立绩效考核系统，并且可以为员工提供更多非正式交流的机会。

二、组织结构的基本类型

根据构成组织结构基本要素的特征，可以大致上将组织结构区分为两类：机械式结构（mechanistic structure）与有机式结构（organic structure）。机械式结构通常控制幅度较窄、高度集权化并且高度正规化，主要采取自上而下的信息交流模式；而有机式结构通常控制幅度较大，结构趋于扁平化，决策过程采取分权化，组织正规化程度较低，拥有全面完整的信息交流模式。机械式结构比较适用于稳定的常规性组织活动的管理。而在瞬息万变的今天，强调信息共享、能够灵活应对环境变化的有机式结构往往更有利于组织的发展。

具体地讲，组织结构主要包括以下四种基本类型：

（一）简单式结构（simple structure）

简单式结构是指正规化程度较低，但却高度集权化、控制幅度较大的“扁平化”组织。在组织建立之初，通常会采用这种结构方式。简单式结构具有灵活性较强、运营成本低、责任明确等优势，比较适用于小型企业组织，但是随着组织规模的扩大，其决策风险会增大，难以应对复杂环境，因此当组织规模发展到一定程度时，往往不再适用。

（二）职能式结构（functional structure）

职能式结构是指根据员工共同的专业知识、经验将其组织在一起的结构。绝大多数组织在其发展的某个时期或阶段都会采用职能式结构。职能式结构有利于培养员工的职业定向，便于管理者进行直接监督，此外专业分工还有益于规模经济的产出，能够减少人员与设备重复配置等浪费现象。但是，职能式结构也存在一些局限性：根据共同兴趣或背景进行分组会加剧不同职能单元之间的分化，不利于各部门之间的协作与交流。此外，还往往容易出现员工过分追求职能目标而忽视组织整体利益的现象。因此，采用职能式结构时，管理者应注意针对这些问题进行必要的干预与协调。

（三）分部式结构（divisional structure）

分部式结构，又称战略经营单位（strategic business units），是指由自我包容的自治单位构筑形成的组织结构，通常组织会根据地理位置、顾客群体或某项产出（产品或服务）来建立分部。与职能式结构相比，分部式结构具有更多自主权，分部经理全面负责本部事务，拥有充分的运营决策权，这有利于实现权责结合、易于绩效评估。此外，分部式结构还有利于包容与适应组织的发展，因此当组织发展到一定规模时，往往会采取这种组织结构设计方法。但是分部式结构也有其局限性，其最大的弊端在于容易造成活动与资源的重复配置，从而导致组织资源利用率降低，总成本上升。此外，各自相对独立、拥有较多自治权的分部之间还容易出现协作不力的现象。

（四）矩阵式结构（matrix structure）

如上所述，在组织结构设计中，职能式结构与分部式结构各有利弊。采用职能式结构，能够有效地整合资源，但却缺乏对产品与市场需求的关注；采用分部式结构，虽然强调以产出为中心，但是却容易造成资源的重复配置与浪费。而矩阵式结构则试图整合这两种组织结构方式的优点，兼顾市场需求与资源优化配置的双重要求。

所谓矩阵式结构是指在传统纵向职能部门系统基础上结合横向产品部门系统形成的组织结构。在矩阵式结构中，通常将职能部门的员工指派到一个或多个项目组中，由项目经理来领导。矩阵式结构创造了双重命令链，员工需接受项目经理与职能经理的双重管理。矩阵式结构若能合理运用，既可以实现资源的优化配置，又能兼顾市场需求。此外，与职能式结构相比，矩阵式结构能够有效加强横向联系与协作，提高项目实施的灵活性，有利于激发员工的创新意识。但是，在矩阵式结构中，由于存在着双重领导，因此有时难以明确责任，而且容易诱发组织冲突，引发权力斗争。为了避免员工应对“一仆二主”状况的尴尬，项目经理与职能经理应经常保持沟通，并注意协调他们对员工提出的各种要求。

在实际组织结构设计中，除了上述四种基本结构形式以外，还普遍存在着融合各种结构要素的所谓“混合式结构”（hybrid structure）。为确保组织的健康发展与效率，无论是哪一种组织结构，都应注意平衡其内部各部门之间的权力分配。

三、组织结构的新发展

随着信息技术的发展与劳动力队伍的变化，组织结构亦有新发展，出现了以下两种典型的新型组织。

（一）网络结构（network structure）

网络结构亦称组合式结构（modular structure），是指由若干组织联合以进行产品开发或提供服务的组织结构。通常在网络结构中存在着一个核心企业，其周围分布着一些卫星组织。核心企业经营其中某些关键业务，并负责协调各卫星组织之间的关系。目前，凭借先进的信息技术，可以在各个组织之间方便地实现基于网络的通讯与协作。网络结构具有极强的灵活性，它使组织不再是一台一成不变的生产机器，而能根据环境变化的要求及时地进行调整。此外，网络结构采取业务外包的形式，有利于资源的整合利用，实现以较少资本推动大规模生产与经营。但网络结构也存在一些潜在的局限性。与传统组织结构相比，核心企业对组织的控制力不足，市场竞争容易导致成本加大，而组织所取得的创新成果也很容易被窃取。

（二）团队型结构（team-based structure）

团队型结构是指主要以团队为基本工作单元来负责完成组织任务的结构模式。在团队型结构中，通常层级较少，正规化程度较低，工作团队本身具有较多的自主权。由于团队型组织赋予员工更多权责，因此能够激发员工的创新性与责任感，有利于增强组织活力。此外，团队型组织还有利于冲破传统层级制度的界限，增进员工之间的交流与协作。但团队型组织的维系与发展需要花费较多的时间和精力，这便对管理者提出了更高的要求。鉴于团队型结构已经成为一种日益普及的组织形式，下一节将专门对其加以介绍。

第三节　组织中的团队

一、团队及其类型

（一）团队的概念

团队（team）是指由两人或多人组成的群体，其成员相互作用、相互影响，为实现特定的共同目标而承担相应责任。团队成员拥有共同目标，他们通过彼此间的相互依赖以及为实现共同目标而进行的合作紧密连接在一起。为了更好地分享信息与协调工作，团队内部需要各种有效的沟通渠道。

（二）团队的类型

组织中存在着多种类型的工作团队，常见的主要包括以下三类：

1．固定团队（permanent team）

固定团队通常负责组织中一组特定的任务。一个部门可能就是一个固定团队，但也存在着非团队取向的部门。在非团队取向的部门中，员工独自工作，并且单独对上级负责。员工们虽然聚集在一起，但他们之间缺乏联系，因此从严格意义上讲不能称之为团队。只有鼓励员工彼此之间不断进行直接信息交流以协调工作进程的部门才是真正意义上的固定团队。

作为团队型组织的基础，自我指导工作团队（self-directed work teams，SDWTs）也属于固定团队。但与团队取向的部门不同，作为工作单元的自我指导工作团队是围绕着工作进程组织起来的，因此小组内部工作高度自治，较少需要上级的管理与监督。此外，自我指导工作团队的成员通常拥有多种互补性而非单一相似性的知识、技能与经验。

2．临时团队（temporary team）

临时团队通常负责执行组织中的一些短期项目或某些决策活动，如公司从不同部门中召集员工来开发一种新产品等。特别任务小组（task forces）属于临时团队，它负责调查某一问题并就此提出决策方案，任务完成后，小组自动解散。所谓的“臭鼬小组”（skunk works）通常也属于临时团队，它往往是以任务为导向，应开发产品或解决难题之需而组建，一般在组织内部具有较高的独立性，较少受规则制度的约束。

3．虚拟团队（virtual team）

虚拟团队是指成员之间凭借电子科技手段跨越时空与组织界限进行沟通的团队。通过先进的电子科技，可以更方便固定团队执行任务，同时也能够超越时空限制组建优秀的临时团队来开发新产品或进行决策。

虚拟团队日益普及与发展的原因主要有以下四个方面：首先，科技的进步促进了虚拟团队的发展。各种新兴的通讯手段（如 E-mail、视频会议、电子聊天室、内联网等）为虚拟团队的组建提供了技术支持。其次，工作任务性质的转变为虚拟团队的发展提供了可能性。传统的基于产品的任务开发通常需要员工集合在一起进行生产活动，而基于知识的工作往往并不要求员工聚集在一起，即使相隔很远，也可以通过信息技术手段进行协作。第三，全球化的发展趋势召唤新的团队形式。随着全球化的发展，跨国公司日益普及，不同地域的集团之间合作与联系越来越密切。而传统的团队形式无法有效地满足这种发展需要。第四，虚拟团队本身的优势是其发展的内因。虚拟团队的超时空特性更有利于知识的共享与使用，同时团队配合的方式可以有效地提高决策的质量。

总之，在组织管理中，团队已发展成为非常重要和普遍的工作单元。团队灵活多样的形式使其无论是在组织的常规工作中，还是执行短期项目时，均能有效发挥作用。此外，凭借先进技术手段组建而成的虚拟团队在全球化发展趋势盛行的今天更是不可或缺的组织形式。优秀的团队在无需上级严格督管的情况下，可以高效协调其内部成员的工作，通过集思广益的合作方式，做出更富有创造性且更加可靠的决策。

但是，在团队中也存在一些潜在的问题。由于团队是需要花费时间与精力去发展和维

护的，因此可能会出现得不偿失的过程损失现象（process losses）。此外，在团队中还容易出现社会惰化现象（social loafing），人们在团队中工作时比单独工作时更不努力。因此，如何加强团队建设，提高团队效能，就成为当代组织管理的首要问题。

二、团队效能及其影响因素

（一）团队效能

团队效能（team effectiveness）是指团队实现其目标、满足其成员需要以及维持自身生存与发展的能力或有效性。衡量工作团队效能的首要标准是其组织目标的实现程度，其次应考虑成员的幸福感和满意度，此外还应考虑团队的生存能力，即维持成员对团队的组织承诺感、确保团队拥有一个优越的发展环境。

（二）团队效能的影响因素

图 16.2 给出了一个团队效能的模型。从中可以看出，影响团队效能的因素主要有三类：

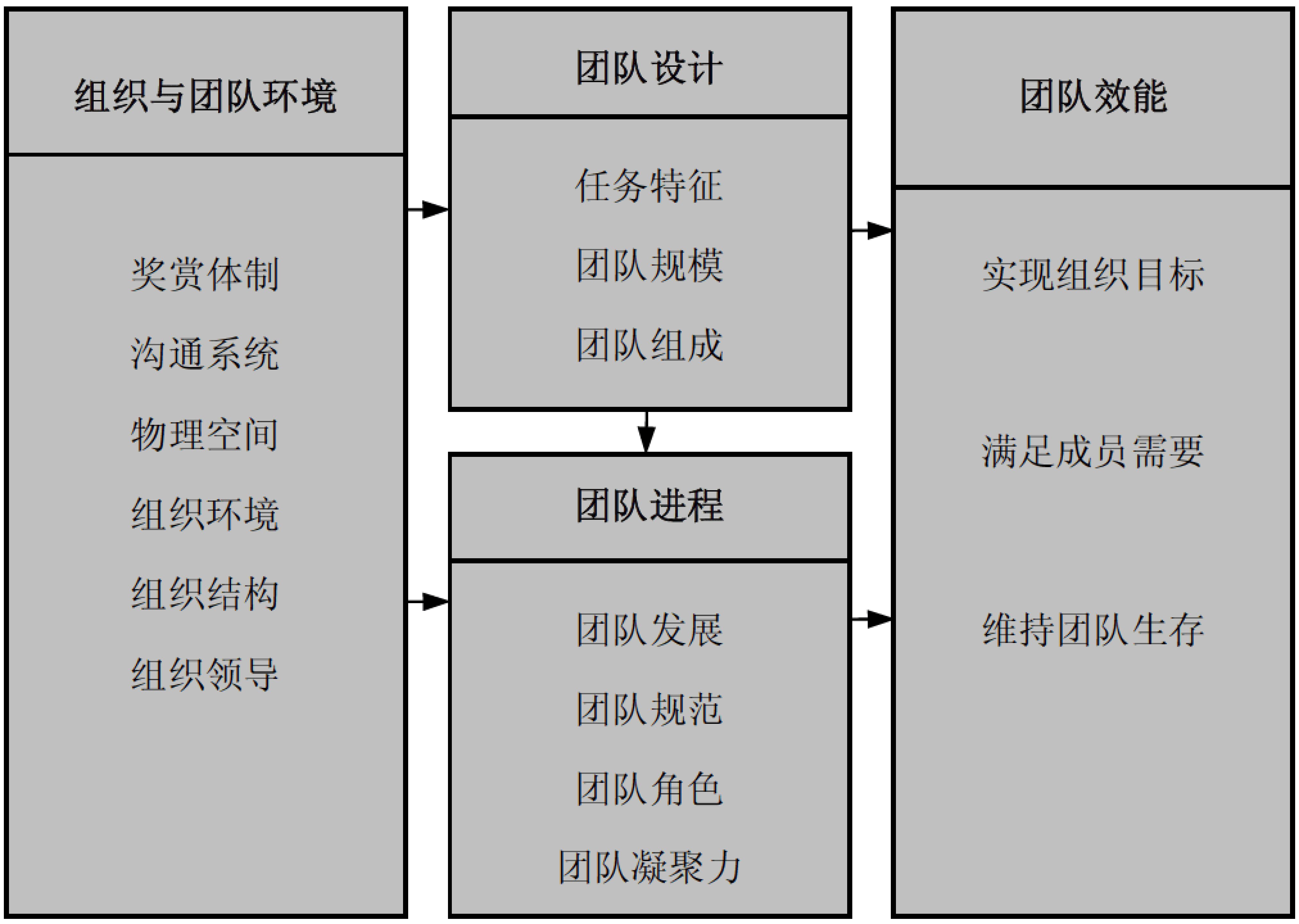


图 16.2　团队效能模型（引自 Steven L. McShane & Mary Ann Von Glinow,2000）

1．组织与团队环境

组织与团队环境因素通过作用于团队设计与团队进程间接影响着团队效能，它主要包括以下六个方面：

（1）奖赏体制

研究发现，将成员对团队的贡献作为论功行赏的标准之一往往更有利于团队的成长与发展。与个人奖励相比，采用固定工资与团队奖金相结合的奖赏体制通常能够更加有效地激发团队的活力。

（2）沟通系统

沟通系统是影响团队成功的重要因素。不良的沟通系统要么会使信息不畅，缺乏反馈；要么会导致信息超载，无从应对。因此，无论是成员聚集在一起进行面对面交流，还是虚拟团队的工作方式，选择和运用恰当的沟通手段都是至关重要的。此外，团队的沟通系统还应与团队的任务和结构相适应。

（3）物理空间

合理布局的物理空间不仅能够促进成员之间的沟通，有时还会直接影响团队完成任务的速度与质量。研究发现，布局合理的物理空间有助于团队成员形成整体意识、对其所属团队产生一种自主感。

（4）组织环境

团队所处的组织环境会直接影响团队的工作。如果组织无法确保必要的资源，就会影响团队的生产力。在竞争性较强的环境下，当成员将团队合作视为保持竞争力的最佳策略时，他们往往能够更加紧密无间地进行合作。

（5）组织结构

鼓励和支持团队成员彼此之间不断进行直接信息交流以有效协调工作进程的组织结构形式有利于提高团队效能。这类组织结构赋予团队更多的自主权，能够增强成员间的沟通与联系。

（6）组织领导

团队需要组织领导者的不断支持。领导者可以通过调整各种环境因素（如奖赏体制、沟通系统、组织结构等）来扶持团队的发展，还可以适时地充当“缓冲器”，保护团队免受各种组织政治手段的侵害。此外，崇尚集体努力的企业价值体系是团队赖以发展的土壤，也需要领导者积极倡导和建立。

2．团队设计

如图 16. 2 所示，团队设计中的结构性特征既会直接影响团队效能，也会通过作用于团队进程间接地影响团队效能。团队设计中的结构性特征主要包括以下三个方面：

（1）任务特征

明确规定的任务有助于团队成员迅速进入角色，而定义模糊的任务则需要耗费团队大量的时间来进行沟通以确定任务分工及方案。除了任务的明确性以外，另一重要的任务特征是任务的互依性。团队非常适合执行互依性较强的任务，因为互依性较强的任务能够激发团队成员一起工作的责任感，而团队配合的工作方式也有利于成员之间的协作。

（2）团队规模

最佳团队规模取决于多种因素。确定团队规模的基本原则是团队应有足数的成员以确保任务的顺利完成，在此前提下，团队规模应尽可能小，以使每个成员都能积极投入团队

活动，保持有效的合作与协调。控制团队规模可以缓解前面提到的社会惰化现象，使每个成员的贡献能够清晰地展现出来。因此，很多成功的企业都采用了小团队协作的工作方式。

（3）团队组成

在团队组建之初，企业就应设法采取各种措施（如挑选具有团队意识的员工、集中培训、建立基于团队的奖赏体制等）以确保员工具备必要的合作动机与能力，使成员意识到合作的重要性和有效性，培养员工的协作能力，如良好的情绪智力及管理冲突的技巧等。

除了关注员工的合作动机与能力以外，团队组成还应考虑其成员的多样性问题。根据团队成员的特征，可以将团队区分为同质性团队（homogeneous team）与异质性团队（hetero geneous team）。同质性团队是指由具备相似技术专长、种族特性、经历或价值观念的成员组成的团队；而异质性团队是指其成员具有多样化的人格特征及背景的团队。同质性团队往往比较适合需要高度协作的任务，而异质性团队则适用于应对复杂的项目或需要创造性解决方案的工作。

3. 团队进程

如图 16. 2 所示，影响团队效能的团队进程主要包括以下四个方面：

（1）团队发展

团队发展需要时间，在此期间，团队成员必须要互相了解、领会各自的角色、发现恰当以及不恰当的行为并学会协作。新成员不断加入和旧成员不断离去，使团队始终处于不断变化的过程之中。如图 16. 3 所示，布鲁斯 • 图克曼（Bruce Tuckman）关于团队发展的五阶段模型形象地描述了团队这种递进演变的过程（图中虚线部分表示随着新成员的加盟或其他因素的影响，团队发展可能会出现暂时倒退的现象）。这五个阶段是：

形成阶段（forming），团队发展的第一个阶段。在这一阶段中，成员尝试着互相了解，衡量作为团体一员的获益和代价。在这个阶段，成员通常会表现得比较有礼貌，乐意遵从现有领导者的权威与规则，并尽力地去适应团队。

爆发阶段（storming），在团队发展的爆发阶段，成员往往会主动出击去担当某种角色，由于竞争经常会出现人际冲突。这是团队发展过程中比较脆弱的阶段。在这一阶段中，团队成员开始慢慢理清各种角色的特定要求，并且开始认同承担角色的人选。

规范阶段（norming），在团队发展的规范阶段，角色确立并且围绕团队目标的意见逐步统一。成员形成了相对一致的心理模型，而这种共同的预期和设想有利于他们之间更加积极有效地进行互动，推动团队进入下一个发展阶段。

运作阶段（performing），在团队发展的运作阶段，其重心由确立和维系关系转变为实现目标。团队强调以任务为中心，为完成任务而努力。在这一阶段中，团队成员已能较好掌握积极进行协作和解决冲突的方法。

停滞阶段（adjourning），由于各种原因，如项目结束、人员调离、成员辞职或被解雇等导致团队解体。

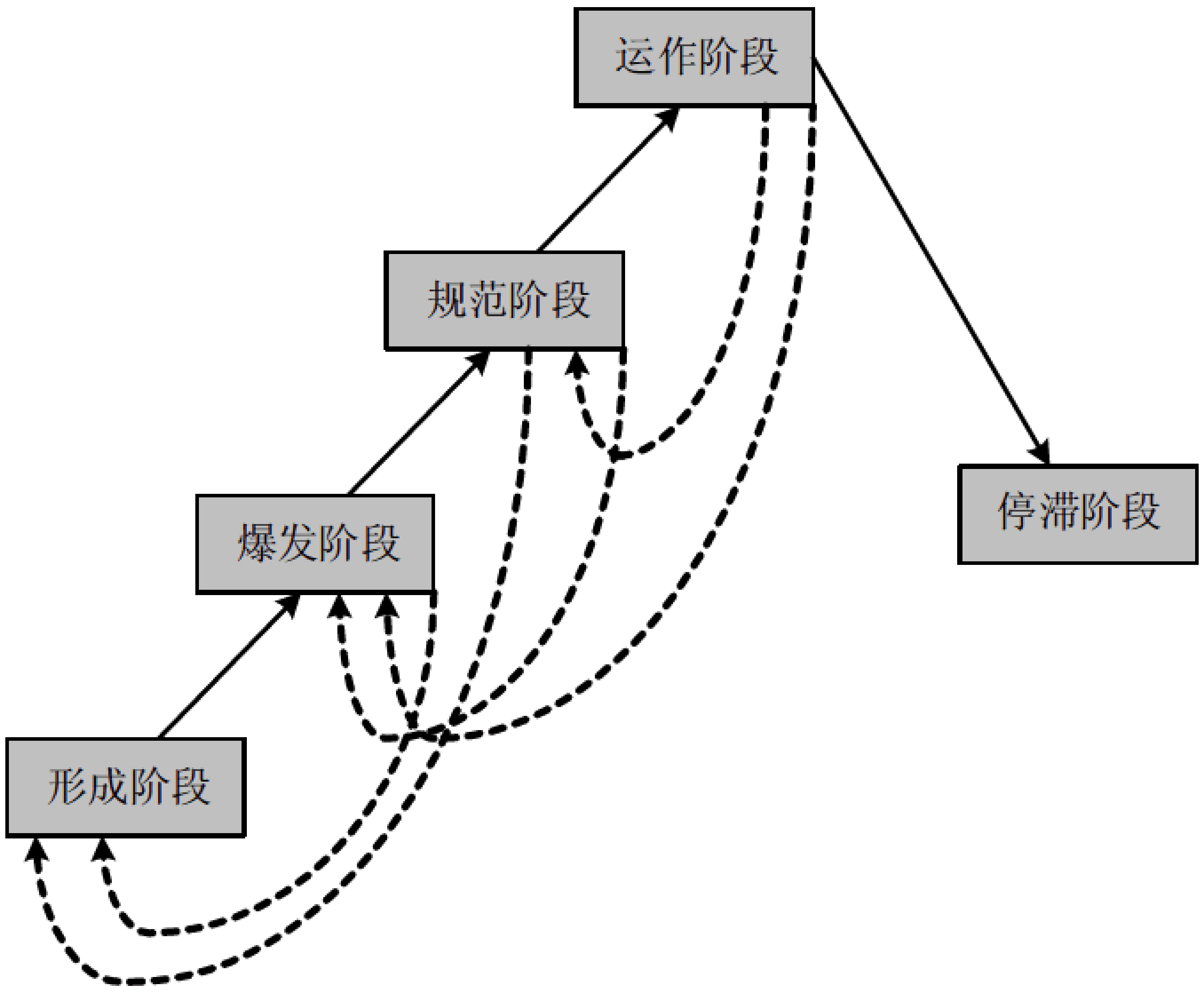


图 16.3　团队发展的五阶段模型（引自 Steven L. McShane & Mary Ann Von Glinow,2000）

（2）团队规范

团队规范是指团队用于约束其成员行为的非正式规则与期望。团队规范通常被用来管理对团队比较重要的成员行为，如共享资源的办法、接待客户的方式等。有些团队规范与组织目标一致，而有些团队规范则与组织目标相冲突。

成员遵守团队规范的动力主要来源于三个方面：一是同事的压力，不符合团队规范的行为往往会遭到同事的抵触甚至排斥；二是高层领导的强化，上级的表扬或奖励会直接强化规范的执行；三是成员对团队的认同，成员往往会自觉遵守通行的规范，因为他们认同团队，希望自己的行为能够与团队所推崇的价值观念保持一致。

影响团队规范形成的因素主要有四个方面：①团队中发生过的重要历史事件：如一些灾难性事故能够加强安全操作规范的执行；②团队形成不久发生的事件：团队成立之初成员之间的交往方式会对他们将来的行为模式产生深远的影响；③团队成员的认知：若团队成员意识到某种行为能够帮助他们更加有效地行使职责，维持团队的生存，则有助于这种行为规范的确立与执行；④团队成员自身的信念与价值观：每一个成员先前的经历及其观念一经带入到团队中，便会影响团队规范的形成与执行。

虽然许多团队规范根深蒂固，但有时却需要矫正和改变。改变团队规范的途径主要有：①以绩效为导向，采取公开讨论的方式筛选并推行有效的团队规范、纠正无益的规范；②挑选能够有效贯彻规范的成员，并积极发挥他们的影响力；③必要时解散并重新组织新的团队。

（3）团队角色

团队角色是指团队中处于某一位置上的人所应履行的一套行为。通常每个团队成员都会具有多种正规角色，这些角色规定着成员应达到的目标以及应负责的事务等内容，引导他们从事各种与任务相关的行为活动。在一定程度上讲，成员完成工作任务的绩效取决于他们对自身角色的认知，即角色知觉。当出现角色模糊、角色间冲突或个人－角色冲突时，

这些正规角色往往会给个体造成压力。

除了各种正规角色外，在团队中还存在着各种成员自愿承担的角色以及随时可以发生互换的非正规角色。无论是哪一类角色，其关键在于成员应能尽职履行角色所承担的责任以确保团队正常运作。

（4）团队凝聚力

团队凝聚力是指团队对其成员的吸引力以及激励他们甘愿成为团队一员的程度。团队凝聚力是影响团队成功与否的重要因素，它像胶水一样粘结着团队成员，以确保每人都能各尽其责。凝聚力更多的是一种情感体验，而不仅仅是利益得失的计量。只有当团队成员真正认同团体的理想和信念，形成强烈的团体归属感时，凝聚力才可成为紧密联结个人与集体的纽带。

任何一个团队要想生存下去，必须具备一定的凝聚力。与低凝聚力的团队相比，凝聚力较强团队的成员往往会利用更多的时间在一起分享信息、进行互动交流。他们彼此体谅，互有好感，人际关系和谐，因此很少会发生冲突。即使有冲突，他们往往也能及时有效地解决问题。在面临压力时，成员之间也能彼此照应、相互支持。由于拥有良好的协作环境，因此当团队规范与组织目标一致时，凝聚力较强的团队通常能够取得较好的任务绩效。

影响团队凝聚力的因素主要包括以下六个方面：

①成员相似性：尤其是在团队发展的早期阶段，具有相似背景的同质性团队较之多样性的异质性团队更容易产生较强的凝聚力。因为与彼此相似的人沟通会强化自身的价值观念，稳定自我同一性。此外，同质性团队成员往往容易就团队目标、实现方法及团队规范等重要问题达成一致，这会进一步增进成员间的信任，减少冲突的发生。但是，在很多情况下，团队需要面对复杂的任务或需要创造性解决问题，这时异质性团队往往更为高效。因此在组织管理中，应注意权衡成员相似性所带来的利弊。

②团队规模：小规模的团队通常更易产生较强的凝聚力，因为人数较少更便于协作，容易达成一致。但这并不意味着人数越少越好，因为人数过少难以保证完成既定的任务。因此，比较有效的方法是在确保任务顺利完成的情况下，尽量精简人员。

③成员的互动：当团队成员直接互动的机会较多时，有助于团队形成较强的凝聚力。直接互动强调面对面的交流，而单凭网络进行远程沟通，会使成员产生离群索居、孤军奋战的感觉，不利于团队产生深厚的凝聚力。

④加盟的难度：如果加盟一个团队稍微有些难度，那么这个团队通常会对其成员产生较强的凝聚力。但是如果难度过大，结果往往会适得其反。适当的加盟难度有助于团队树立威望，增强团队的吸引力。与此同时，由于自身也有过类似的经历，团队中的老员工往往也会乐于帮助那些刚刚“过五关、斩六将”的新员工。

⑤团队的成功：随着团队不断取得成功，其凝聚力会得以增强。成员在团队的成功中可以获得更强的社会同一性，并且增强对团队未来发展的信心。因此，向成员宣布并庆祝团队的胜利有助于振奋人心，提高团队的凝聚力。

⑥外部竞争与挑战：当面对外部竞争或具有挑战性的任务时，团队的凝聚力会得到加强。这既包括来自于外部竞争对手的威胁，也包括与组织内部其他团队之间进行的“友谊赛”。总之，各种竞争与挑战能够使团队成员紧密团结在一起，寻求协作与社会支持，以实

现团队目标。当然，外部竞争也不宜过盛。研究发现，若竞争压力过大，虽然团队凝聚力在提升，但团队的工作效率仍会有所下降。

三、团队建设

团队建设是旨在增进团队发展与改善团队机能的各种正式活动。它既适用于新建团队，也适用于在团队发展过程中出现了倒退现象的老团队。优质的团队建设活动能够促进团队发展，增强团队凝聚力。

（一）团队建设活动

常见的团队建设活动包括以下四种：

1．角色定义

角色定义的主要目的是检查团队成员之间的角色期待，并使他们明确将来各自应履行的角色责任。通常，团队成员先各自描述一下对自身角色的认知以及对他人角色的期待，然后基于这种角色知觉的讨论，成员修正并完善各自的角色。

2．人际互动

人际互动的主要目的是通过消除潜藏的误解使团队成员之间建立真诚互信的沟通环境。人际互动活动通常会为团队设计一个特殊的挑战（如野外探险等），通过让成员在陌生情境下解决各种问题来了解彼此的长处和缺点，发现良好人际关系对于他们潜能发挥的重要作用，使他们深刻体会到信任和尊重对团队协作的必要性。

此外，还可以通过对话的方式增进团队成员间的相互了解，通过团队成员之间正式或非正式的交谈来加强沟通，在了解各自基本设想与心理模型的基础上建构团队共同的思维模式。

3．目标设置

目标设置的主要目的是使团队成员明确团队的绩效目标，增强团队实现目标的动机，并就团队成员的操作绩效建立有效的反馈机制。研究发现，目标设置是团队建设中很重要的一个维度。

4．问题解决

问题解决的主要目的是检查团队中与任务相关的各种决策活动，并设法使其更有效。有些团队通过参加一些模拟游戏来提高解决问题的技巧，在假设的情境下进行各种决策活动，不仅有助于团队成员提高决策本领，还可以完善人际互动过程。

（二）团队建设应注意的问题

随着团队在组织中的普及，团队建设活动也越来越普遍。但是如果应用不当，团队建设活动将无法体现其应有的价值。因此，进行团队建设时还应注意以下四个方面的问题：

1．对症下药

开展团队建设活动的误区之一是预先没有把握好团队的真正需要。上述四种团队建设活动服务于不同的目的，并没有适合所有团队所有情况的灵丹妙药。因此，在组织团队建

设活动时，必须考虑团队的类型与需要。

2．不断强化

团队建设活动不是“预防疫苗”，不能一蹴而就。团队建设是伴随团队成长和发展的一个不断前进的过程，一些训练项目之所以失败就是因为缺乏必要的后续措施，无法持续发挥项目的作用。

3．学以致用

团队建设活动的宗旨是服务于团队发展。因此，应鼓励团队成员在实际工作中积极进行反思，将在各种训练活动中获得的体会和领悟有机地融入到工作中去，为团队发展贡献力量。

4．寻求平衡

团队建设活动可能会导致其成员过分地效忠所属团队。这种倾向虽然能够有效地增强团队内部的凝聚力，但同时也会成为诱发团队与组织中其他部门之间发生冲突的潜在因素。

本章小结：

1．组织是指人们为实现特定目标相互依赖、彼此协作而形成的机构或团体。它是社会生活的基本单元，是联系个人和社会的桥梁和纽带。

2．组织结构是指组织内部的工作分配以及指导组织活动的协调、沟通、工作流程和正式权力的模式。形成组织结构的两个必要条件是工作分配与工作协调。

3．组织结构主要包括控制幅度、集权化、正规化与部门化四个基本要素，其基本类型主要有简单式结构、职能式结构、分部式结构与矩阵式结构。随着信息技术的发展与劳动力队伍的变化，出现了网络结构、团队型结构等新型组织。

4．团队是指由两人或多人组成的群体，其成员相互作用、相互影响，为实现共同目标而承担相应责任。组织中常见的工作团队有固定团队、临时团队和虚拟团队。

5．团队效能是指团队实现其目标、满足其成员需要以及维持自身生存与发展的能力或有效性。影响团队效能的因素有：组织与团队环境、团队设计和团队进程。

6．组织与团队环境因素通过作用于团队设计与团队进程间接影响着团队效能，它主要包括奖赏体制、沟通系统、物理空间、组织环境、组织结构和组织领导。

7．团队设计中的结构性特征既会直接影响团队效能，也会通过作用于团队进程间接地影响团队效能。团队设计中的结构性特征主要包括任务特征、团队规模和团队组成。

8．同质性团队是指由具备相似技术专长、种族特性、经历或价值观念的成员组成的团队，它往往比较适合需要高度协作的任务；异质性团队是指其成员具有多样化的人格特征及背景的团队，通常适用于应对复杂的项目或需要创造性解决方案的工作。

9．影响团队效能的团队进程主要有团队发展、团队规范、团队角色和团队凝聚力。团队发展大致经历形成、爆发、规范、运作和停滞五个阶段。团队规范是指团队用于约束其成员行为的非正式规则与期望。成员遵守团队规范的动力主要来源于同事的压力、高层领

导的强化以及成员对团队的认同。影响团队规范形成的因素主要有团队中发生过的重要历史事件、团队形成不久发生的事件、团队成员的认知以及团队成员自身的信念与价值观。团队角色是指团队中处于某一位置上的人所应履行的一套行为。团队中有各种正规角色、成员自愿承担的角色以及随时可以发生互换的非正规角色。

10．团队凝聚力是指团队对其成员的吸引力以及激励他们甘愿成为团队一员的程度。影响团队凝聚力的因素主要有成员相似性、团队规模、成员的互动、加盟的难度、团队的成功以及外部竞争与挑战。

11．团队建设是指旨在增进团队发展与改善团队机能的各种正式活动。常见的团队建设活动主要有角色定义、人际互动、目标设置以及问题解决。进行团队建设应注意对症下药、不断强化、学以致用、寻求平衡。

本章思考题：

1．什么是组织？其特点和功能有哪些？为什么要研究组织？
2．什么是组织结构？试析构成组织结构的两个必要条件。
3．简析组织结构的四个基本要素。
4．试比较组织结构的四种基本类型，并分析组织结构发展的新趋势。
5．什么是团队？试比较团队的三种类型。
6．分析虚拟团队日益普及的原因及其优缺点。
7．团队效能体现在哪些方面？试分析团队效能的模型。
8．分析团队设计中的结构性特征及团队发展的五个阶段。
9．什么是团队规范？其作用是什么？如何形成团队规范？
10．什么是团队凝聚力？其作用是什么？如何增强团队凝聚力？
11．结合管理实践，谈谈团队建设应该注意的问题。

第十七章　组织气氛和士气

科学的组织结构和优秀的工作团队是组织发挥效能的物质前提和基础。但是仅有这些是远远不够的，还需要尽可能营造良好的组织气氛和士气。组织气氛和士气虽然看不见、摸不着，但却真真切切地影响着组织或组织中的每个个体。如果说组织结构是组织的硬件，那么组织气氛和士气则是组织的软件。本章拟对影响组织气氛和士气的几个重要因素进行详细分析。

第一节　组织文化

一、组织文化的概念及构成要素

组织文化（organizational culture）是指支配组织成员在面对问题和机遇时进行思考和行动的共同设想、价值观及信念的基本模式。它是规范组织行为的一种无形力量，是影响组织气氛和士气的重要因素。如图 17.1 所示，这些共同设想属于组织文化中最深层的部分，是组织成员思考和行动赖以产生的基本前提和假设，它们常常会以无意识的方式对人们产生不容置疑的影响；信念代表着人们对于现实的认知和领悟；而价值观则是指比较稳定和持久的态度，它有助于人们对各种思想和行为进行价值判断。总之，作为组织文化的基本要素，共同设想、价值观、信念通常潜伏在外显的组织行为背后，虽然不容易被直接观察，但它们的影响无处不在。

为了理解组织文化的内涵，通常需要解读其人为产物（artifacts），即通过表征组织文化的具体符号和标志来间接地理解其本质。如图 17.1 所示，常用于解读组织文化的人为产物主要有组织语言、组织轶闻与传奇、组织仪式与典礼、组织的物质结构等。组织语言包括组织内部的流行语、称谓方式、领导者的口号与比喻说法等。组织轶闻与传奇记录着组织成员在某一特定情境下的具体反应，可以为其他组织成员充当示范榜样或反面教材。当组织轶闻与传奇真实可信，并且在组织内部广为流传时，能够有效地发挥传播组织文化的功能。组织仪式是指生动体现组织文化的日常组织生活的常规和惯例，而组织典礼则是指正式的仪式，如为表彰优秀员工举行的表彰会以及为庆祝新产品发布举行的庆功会等。组织的物质结构包括组织建筑的外观和风格、内部装饰和布局及各种工作用品如服装、信封等标志。通过这些人为产物，可以感知、理解和解读组织文化。

通常所说的组织文化是指被绝大多数组织成员接受的主流文化（dominant culture）。相对于这种主流文化，在组织的不同部门、区域和群体中还存在着各自相对独立的亚文化

（subculture）。有些亚文化支持主流文化所倡导的核心价值观；而有些亚文化则与主流文化背道而驰，这种亚文化通常被称为反文化（counterculture）。反文化容易在员工中造成混乱和迷惘，甚至引起矛盾和冲突，但正是由于它的存在，使组织的主流规则受到质疑和监督，经常被评判或评估，这在一定条件下能够为组织文化的发展提供动因。

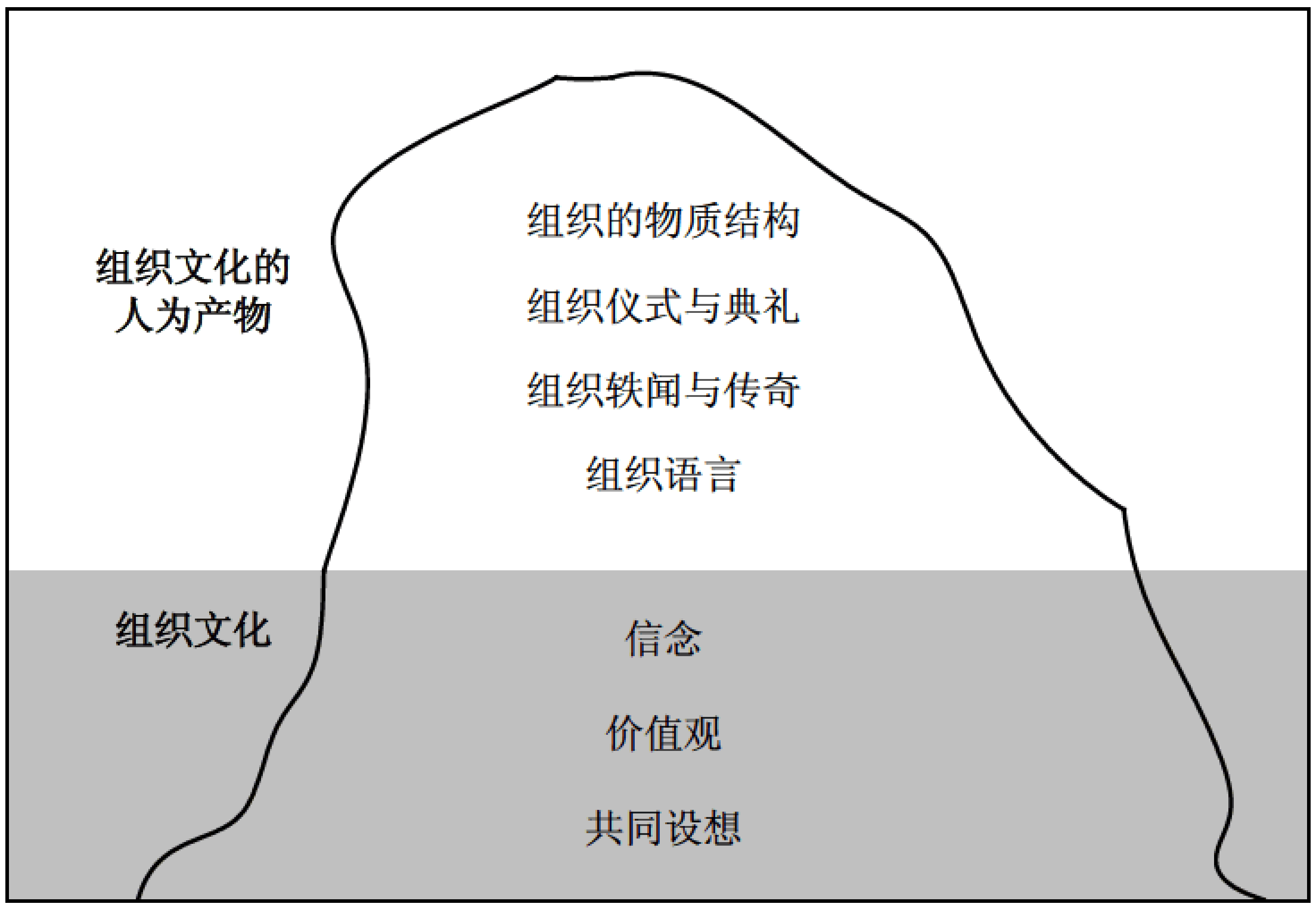


图 17.1　组织文化的构成要素（引自 Steven L. McShane & Mary Ann Von Glinow,2000）

二、组织文化的功能

文化中蕴涵着力量，创造并维持一种适宜的组织文化是许多企业成功的秘诀。组织文化的功能主要表现在以下五个方面：

（一）导向功能

组织文化的导向功能体现在两个方面：一是引导组织成员的思想和行动；二是指导企业整体的价值取向和行为。

（二）约束功能

作为一种深入人心的社会控制形式，组织文化深刻地影响着组织成员的决策与行为。

（三）凝聚功能

共有的设想、价值观及信念将组织成员联结在一起，形成团结一致的整体。

（四）激励功能

组织文化激发组织成员从内心产生一种高昂情绪和奋发进取的精神。

（五）辐射功能

良好的组织文化不仅对组织成员产生影响，而且通过各种渠道向社会传播和扩散。

什么样的组织文化才能发挥上述五种功能呢？研究者归纳出了组织文化三方面的特征，并分析了它们对组织文化作用的影响。这三方面的特征分别是：（1）组织文化的强度。当主流价值观念深入人心且长期被绝大多数成员支持和接受时，那么可以称其为强文化（strong culture）；相反，若主流文化持续时间较短、尚未被多数人所接受，只是管理层少数人倡导和支持，那么就可称其为弱文化（weak culture）。强文化对人的影响大，但其影响是好是坏、是否有利于促进组织发展，受制于它的第二个特征，即文化的匹配性。（2）组织文化与组织环境的匹配性。只有当组织文化的内容与组织所处的环境相匹配时，强文化才能够有效地促进组织绩效和组织发展。如果是一种错误的或与组织环境不匹配的文化，那么它不仅不能促进组织发展，还会阻碍组织的发展。（3）组织文化的适应性。任何组织文化都不是一成不变的，应该具有第三个特征即适应性，因为变化是组织生存和发展中必要且必然的一部分。只有不断关注环境中发生的变化并且鼓励创新以紧跟变化节奏的适应性文化，才会更有利于组织发展。

三、组织文化的冲撞与融合

在组织合并和重组过程中，预先考察两个或多个企业在组织文化上的相容性，在此基础上再采取有效实现组织文化融合的策略，对于组织合并或重组的质量及合并或重组后的管理至关重要。核心价值观截然对立的企业合并后往往会发生组织文化的冲撞，影响组织发展，因此有必要采取一定的措施来尽量减少文化冲撞带来的负面影响。研究者们在大量研究和总结实践经验的基础上，提出了实现组织文化融合的四种策略。

（一）同化（assimilation）

同化是指收购方利用自身强大的文化优势消化被收购方文化，使被收购方员工自愿接受收购方的企业文化价值观。很显然，当被收购方原有的组织文化较弱，而收购方的组织文化比较强大且目标指向明确时，比较适合采取、实施同化策略。通过同化来实现组织文化融合，不太会发生剧烈的文化冲撞，因为被收购方原有的文化不会产生太多阻碍，而员工本身也希望能够找到一种更好的选择。

（二）去文化（deculturation）

去文化是指收购方通过将自身的文化价值观强加于被收购方员工来实现组织文化的融合。若被收购方原有的组织文化严重阻碍合并后的企业发展，则有必要采取去文化的策略。但去文化策略往往难以奏效，有时甚至会适得其反，因为这种文化入侵的方式容易遭到被收购方员工的抵制，从而严重延缓和干扰企业的并购过程。

（三）整合（integration）

整合是指联合两种组织文化，将其合成为一种崭新的、兼具先前各自优势的复合型文化。当双方原有的文化特征具有一定的共通性，且双方都意识到各自现有的文化本身存在一些问题，愿意采纳一套新的、更加有效的文化价值观时，整合策略最为有效。但是整合

的过程比较缓慢，而且由于保守势力的存在，往往危机四伏。

（四）分离（separation）

分离是指合并双方达成一致，在文化或组织活动方面保持各自相对的独立性。当双方行业不同，而不同行业所要求的最适宜的文化价值观也不尽相同时，比较适合采用分离策略。然而，由于收购方总想控制决策权，因此很少有被收购方能够长期保持不受干扰的独立状态。

四、组织文化的巩固与发展

无论是实现组织文化的融合，还是重塑组织现有的文化价值观，组织领导者都必须掌握巩固与发展组织文化的策略。如前所述，诸如组织语言、组织轶闻与传奇、组织仪式与典礼、组织的物质结构等人为产物能够有效地传播组织文化，因此在改变组织文化的过程中，应注意充分发挥人为产物的表征作用，通过消除旧的人为产物、引进和推行新的文化象征来巩固、发展组织文化。针对组织文化的巩固与强化问题，研究者们总结了五种有效的方法。

（一）通过创始人与继任领导者的行为活动

创始人是组织文化的开创者，他们在创立组织的同时也架构了支持其个人文化价值观的体系，创始人为组织打造的文化烙印往往会在相当长的时间里伴随着组织的成长与发展。除此之外，继任领导者的行为活动同样也会影响组织文化的巩固与发展。变革型领导通过传达和实践他们对于未来的构想来进一步巩固和发展组织文化，这不仅仅需要口头鼓吹，更需要倾力而为。

（二）推行与组织文化一致的奖赏体制

具有不同组织文化的企业往往会采取不同的奖赏体制，而与组织文化一致的奖赏体制能够有效地巩固企业的组织文化。例如，为了发扬企业员工的主人翁精神，采用股票奖励的方法来表彰优秀员工，通过让员工持股，使他们感受到自己是企业的一份子，个人利益与企业利益休戚相关，这样就可以进一步强化企业所倡导的组织文化精神。

（三）维持稳定的员工队伍

组织文化植根于每个员工的头脑中，组织轶闻通常是口耳相传，很少被正式记录下来，而在正规的程序手册中也很难找到组织仪式的具体规程。因此，维持相对稳定的员工队伍有助于传播和巩固组织文化。若组织内部大规模裁员或员工大量流失，则无形之中会削弱组织文化的力量；相反地，若组织在短期内迅速扩充人员或兼并大量企业，由于新进员工需要时间来理解和接受企业的组织文化，因此也不利于组织文化的巩固。鉴于此，可以通过适当调节员工增长率、解决人员流失问题来保持已有的组织文化。

（四）建立有效的文化传播网络

组织文化是通过学习不断获得的，因此建立有效的文化传播网络对于巩固和发展组织文化是非常必要的。传播组织文化的渠道主要包括正式传播途径（如公司的例会、内部刊物、企业内联网等）和非正式传播途径（如传播小道消息、员工私下交换意见等）。管理者应积极进入各种传播网络，创造机会与员工分享组织轶闻，实践组织仪式，宣扬企业精神。

（五）挑选员工并对其进行组织社会化

挑选与企业组织文化相匹配的员工并对其进行组织社会化，也是巩固组织文化的有效途径之一。良好的个人—组织匹配度有利于员工对组织环境的适应和接纳，同时也可以提高员工的工作满意度和组织忠诚度。组织社会化是指组织成员领悟胜任本职工作所必需的价值观念、行为表现以及社会知识的过程。理解并悦纳企业文化所倡导的价值观是组织社会化活动的重要组成部分，它不仅影响员工将来的职业发展，也会影响到企业组织文化的巩固与稳定。

第二节　组织权力

一、组织权力的概念

组织权力（organizational power）是指个体、团队或组织所拥有的影响他人的潜在能量。权力并非改变他人态度或行为的具体行动，而是可以实施这些行动的潜在力量。它是组织发挥作用、行使职能的重要方式和根本保证，是影响组织气氛和士气的重要因素。在很多情况下，拥有某种权力的人不知道自己拥有这种力量，因而并未行使自己的权利；或者受对方影响的一方也可能不知道对方拥有的力量，因而没有表现出敬畏或服从的态度。所以，权力产生的前提是一方或双方意识到对方的存在，特别是一方意识到他有赖于另一方提供某种有价值的资源。

由于专门化分工的普遍性和资源的有限性，使得组织中遍布着各种相互依存的权力关系，所以组织权力可以说是组织与生俱来的一部分。在权力关系中，除了强势方会对弱势方产生影响外，弱势方同样会对强势方产生一种反权力作用（counterpower）。只有在权力和反权力的共同作用下，才能有效地维系双方的互动。例如，在图 17.2 中，个人 A 对个人 B 拥有权力，属于权力关系的强势方，因为他不仅可以通过人际互动影响个人 B，还可以直接干预个人 B 目标的实现，而个人 B 虽然不能干预个人 A 目标的实现，却也可以通过人际互动实现自己的反权力作用。

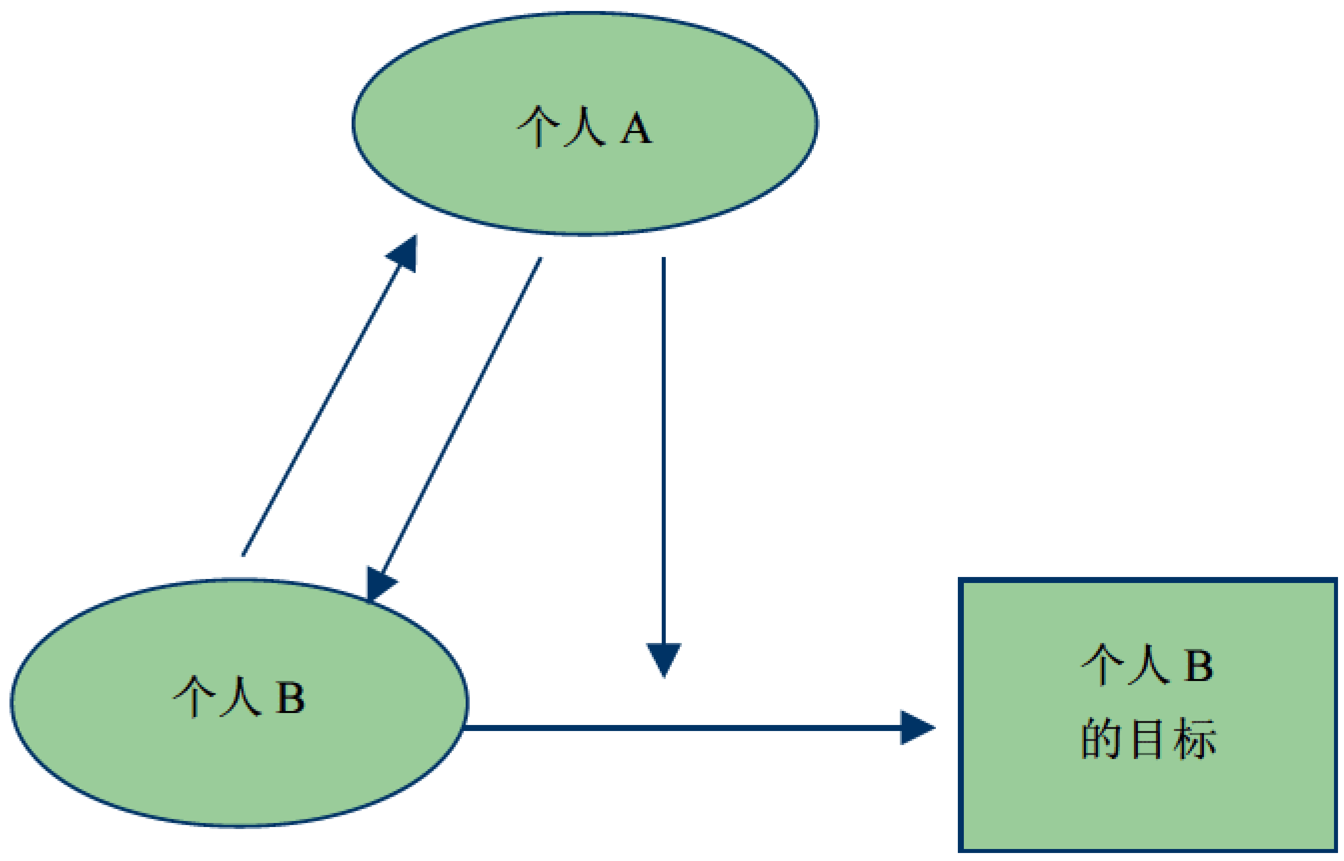


图 17.2　权力关系模型（引自 Steven L. McShane & Mary Ann Von Glinow,2000）

二、组织权力的来源和影响因素

图 17.3 给出了一个组织权力产生机制的模型，从中可以看出权力的不同来源及影响权力产生作用的一些因素。

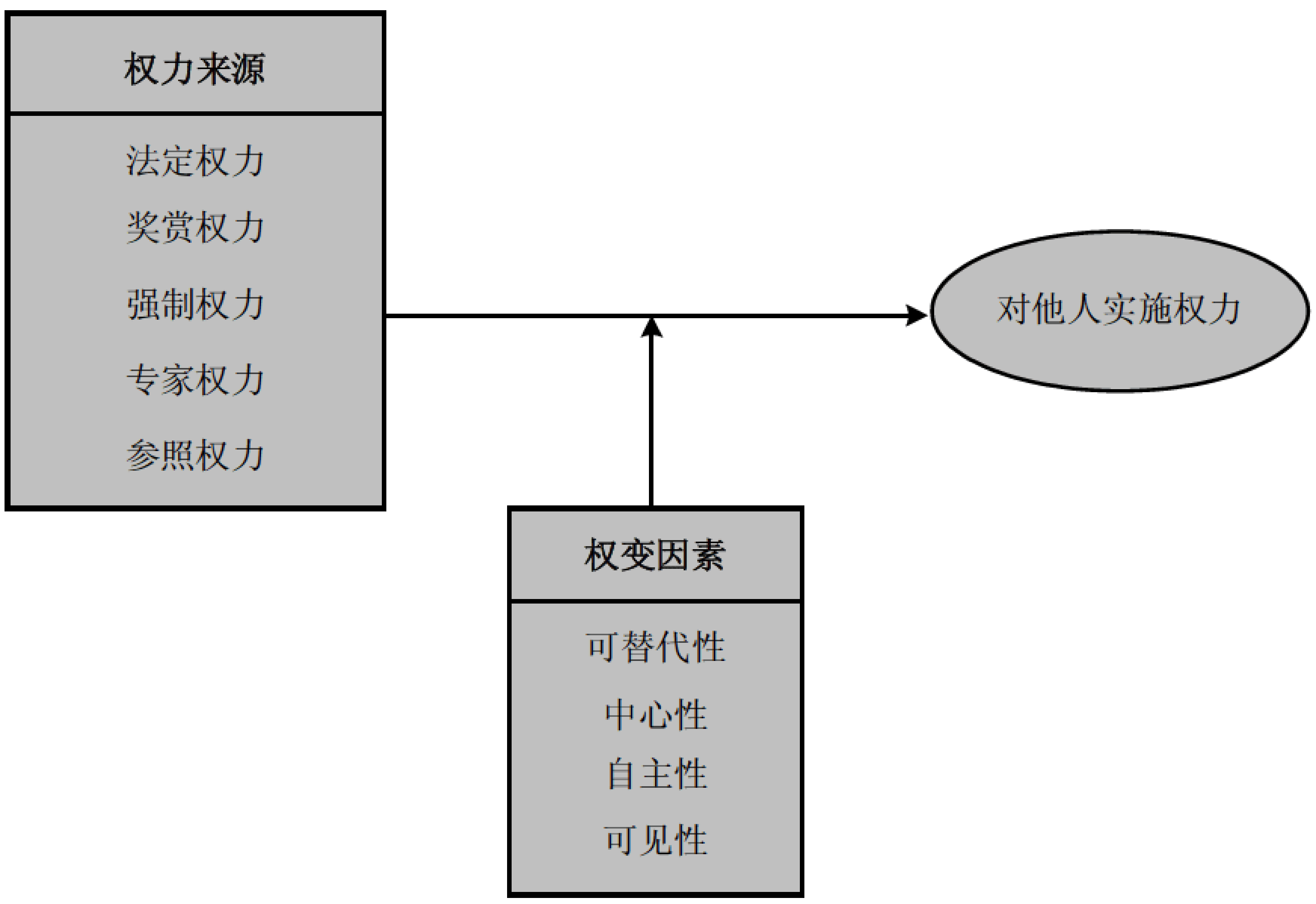


图 17.3　组织权力产生机制模型（引自 Steven L. McShane & Mary Ann Von Glinow,2000）

（一）组织权力的来源

1．法定权力（legitimate power）

法定权力是指通过正式的职权来影响他人的能力。法定权力基于个体在组织中的职位，但是它的有效行使还有赖于组织成员的认同程度。研究发现，权力距离（power distance）是影响组织成员服从行为的重要因素。所谓权力距离，是指组织成员对权力分配不均的接受程度。在高权力距离文化背景下，组织成员倾向于服从。此时，接受命令的人往往缺乏判断与分析，只是盲目地听从指令。

2．奖赏权力（reward power）

奖赏权力是指通过控制给予他人奖赏或消除惩罚来影响他人的能力。在组织中，上级可以通过控制加薪、升职、休假等机会的分配来对下属实施奖赏权力，而下属也可以通过组织的正当渠道（如 360 度反馈系统）实现对上级的奖赏权力。

3．强制权力（coercive power）

强制权力是指通过施加惩罚来影响他人的能力。管理者可以通过训斥、降职甚至解雇等方式对组织成员行使强制权力。团队中的同伴压力有时也有助于发挥强制权力的作用，研究发现，借助于团队成员之间的相互监督能够有效实现工作激励，提高团队成员的出勤率和绩效。

4．专家权力（expert power）

专家权力是指凭借专业知识和技能发挥影响的能力。在知识经济日益发展的今天，拥有专家权力的普通员工会越来越多，他们将专业知识转化为生产力，影响着组织目标的实现。

5．参照权力（referent power）

参照权力是指凭借他人的认同和尊重发挥影响的能力。参照权力主要基于个人魅力，因此形成的过程比较缓慢，但是它一旦形成，就会导致追随者极大的认同和信任。

在上述五种组织权力的来源中，法定权力、奖赏权力和强制权力主要基于权力拥有者的职位，而专家权力和参照权力则源于权力拥有者自身的特征和素质。这五种权力所产生的影响各有不同：强制权力容易引发抵触和不信任；奖赏权力和法定权力常使人产生被动的、缺乏热情的服从；而专家权力和参照权力则会使人产生由衷的承诺，从内心里认同权力拥有者的要求，心甘情愿地去执行。

除了上述五种权力来源以外，随着组织结构的扁平化及信息传递速度的加快，信息在组织权力中发挥着越来越重要的作用，有研究者甚至将其单列为一种组织权力的来源，即信息权力（information power）。信息权力主要有两类：一种是通过控制信息的流通和解释来影响他人的能力；另一种是通过有效应对组织中的不确定性来发挥影响的能力，主要表现为对环境变化的预防、预测及包容。

（二）影响组织权力发挥作用的权变因素

如上所述，组织权力的来源影响其作用的发挥。除此之外，权力本身的一些权变特征也会影响权力作用的大小。

1．可替代性（substitutability）

可替代性就是权力拥有者被他人替代的可能性，或者说是指权力的受众选择余地的大小。当权力拥有者垄断资源时，其权力最大。为了提高自身的不可替代性以增强权力，权力拥有者常常会采取各种手段来表明或暗示自己拥有独一无二的资源。

2．中心性（centrality）

中心性是指权力拥有者和他人相互依赖的程度。中心性有两个维度：一是权力影响的范围；二是权力影响的速度。随着中心性的增加，权力的影响力会增强。

3．自主性（discretion）

自主性是指无需参照具体规则或征得他人许可进行决策的自由度。缺乏自主性会大大削弱权力的行使力度和速度，甚至使权力名存实亡。一些管理者甚至是领导者在各种规则和要求面前感到束手无策，就是缺乏自主性的表现。

4．可见性（visibility）

可见性是指权力及其来源为他人知觉和意识到的程度。只有当自身的权力或其基础为人所知时，才能有效地发挥作用。因此，将自己拥有的权力公布于众，有效行使权力，扩大交际范围和频率，将自己的优势和特长充分展现出来，通过各种线索（如文凭、荣誉称号、衣着打扮、办公室的规模及布置等）传达自己拥有某种资源的信息等，都可以达到提高权力可见性的目的。

总之，各种权变因素会放大或缩小权力的影响力。这种影响力可能会导致抵触和不信任，也可能会令人做出被动的服从，还有可能使人做出由衷的承诺。权力的影响力不仅局限于对他人的影响，权力拥有者本身也会受其影响。如果没有正确的引导，不加节制地追逐或滥用权力，将会导致各种不良组织政治行为的滋长与蔓延。

第三节　组织政治

一、组织政治的含义及利弊

组织政治是指通过一定手段来影响他人以达到个人目的的活动，是组织根深蒂固的一部分，对组织气氛和士气具有重要影响。在多数情况下，尤其是在以牺牲他人或组织利益为代价攫取个人权力时，组织政治无益于甚至有害于组织发展。为了获取权力而从事大量的组织政治活动往往会浪费工作时间，干扰正常的工作秩序和进程。同时过多的组织政治行为还会降低组织成员之间的信任感，阻碍组织沟通，损害组织凝聚力。在充斥着组织政治活动的团队中，员工经常处于应激状态，工作热情下降，人员流动增多。然而，并非所有的组织政治行为都有损于组织发展，例如，员工在工作环境中注意自己的衣着打扮、仪容仪表和言谈举止，与上司打交道时注意方式方法，采取一定的手段宣传自己的主张和建议等，尽管也是为了达到个人的目的，却可能有益于组织的管理和发展。有人指出，组织

政治是组织变革的前提条件和重要组成部分。

那么，究竟应该如何评判组织政治行为呢？一般应考虑以下三个基本原则：（1）实用原则，即考虑该政治策略或行为是否能够为组织或组织中的大多数人带来最大利益；（2）权利保障原则，即考虑该政治策略或行为是否侵犯了他人的合法权利，是否违背道德规范；（3）公平分配原则，即考虑该政治策略或行为是否能够实现公正无偏的利益分配。

二、组织政治行为的类型或表现

研究者们对组织环境中的各种政治行为进行了分类，主要包括以下六个方面：

1．攻击或责备他人

攻击或责备他人是最常见、最直接的组织政治手段，通过贬低他人，将问题的责任归咎于他人，为自己的过失寻找借口，以维护自身的形象，体现自身的优点和长处。

2．控制信息

信息不仅是组织权力的来源，而且也可以充当组织政治的工具。这主要表现在：通过控制信息的流通和解释来限制对手的工作，增强自身的权力基础；通过操纵资料来突出自身的优点，掩饰不足；通过秘藏资料信息来强化自身的不可替代性，提高权力的影响力；通过控制信息沟通的渠道来维护自身的地位和利益。

3．形成联盟或帮派

所谓联盟或帮派（coalition），是指试图通过组内成员资源共用、权力共享的方式来影响他人的非正式团体。当就某个共同目标达成一致，且单独行动无法实现目标时，容易结成联盟或帮派。通过数量上的优势和资源上的集结，联盟可以实现单人无法独立实现的目标，并且可以吸引关注，扩大影响。虽然形成联盟或帮派有时是基于良好的意愿，但通常是为了实现少数人的目标，而且无形之中会给他人带来压力。

4．培植网络，积聚人气

通过培植人际关系网来实现自己的目标，也是常见的组织政治手段。可靠的人际网能够帮助人们获得有价值的信息，增强自身的专家权力基础；广泛的人际网有助于培养同盟和支持者，为项目的申请和实施提供帮助；稳定的人际网还有助于成员增强自身的参照权力，提高威信和声望。

5．施恩图报

以德报德是人际互动的基本准则。通常人们会对曾经帮助过自己的人心存感激，在他们需要帮助的时候会更乐意去成人之美。因此，通过小恩小惠等方式拉拢人心，以图对方在必要的时候给自己以更大的回报，也是一种常见的组织政治手段。

6．印象管理

印象管理（impression management）是指努力打造正面公众形象的一系列活动。当有意识地通过印象管理来迎合他人的要求以达到自己的目的时，印象管理可被视作一种组织政治手段。

三、影响组织政治活动的条件和因素

组织政治行为在适当的条件下会更加活跃，得以不断滋长和蔓延。影响组织政治活动的条件和因素有以下两个方面：

（一）组织环境因素

当组织中资源不足或资源分配的规则比较模糊或复杂，甚至根本无章可循时，员工倾向于使用组织政治手段，以确保自身利益免受侵害；当组织处于权力之争的特殊时期（如领导班子换届、改选、并购、重组等）时，组织政治活动更加活跃；当组织对待各种组织政治行为的态度比较宽容甚至纵容时，会助长各种不良的组织政治行为。

（二）个体特征

影响组织政治行为的个体特征包括人格因素和性别因素。

人格因素主要表现为：有强烈权力欲望的个体倾向于更多地采用组织政治手段；与外控者相比，由于内控者相信自己是命运的主宰，因此在适当的情境下，更容易参与组织政治活动。信奉马基雅维利主义[①]的人否定政治活动与道德有关，认为诡计和欺骗在追求和维护政治权力的过程中是完全正当的，因此他们更喜欢运用各种组织政治手段来达到自己的目的。

组织政治行为的性别差异主要表现为：男性具有较强的自尊感和防御意识，因此喜欢采用有利于维护自身地位和权力的交际风格与行为方式。他们更倾向于采用直接的印象管理方法来提升自己的形象，一般不会主动认错，而会推卸责任，转而责备他人。与之相比，女性则不太善于使用各种组织政治手段，通常愿意主动承认错误，倾向于采用组建联盟和积聚人脉等比较温和的方式来实现自己的目的。

四、组织政治的管理措施

为了防止不良组织政治行为的泛滥，管理者可采取一些必要的应对措施，主要包括以下八个方面：

（1）尽可能保证充足的关键资源。

（2）资源稀缺时，制定资源获得和利用的明确而具体的规则，杜绝暗箱操作。

（3）建立畅通无阻的信息交流系统。

（4）组织内部发生变动时，应加强与员工的沟通，增加透明度。

（5）必要时重构团队和组织规范，以清除组织政治行为对组织目标的干扰。

（6）正确用人，避免权谋者当政。

[①] 马基雅维利：意大利文艺复兴时期的政治思想家，历史学家，剧作家，近代西方政治学的奠基人。他所主张的政治权术思想被后人称为马基雅维利主义。

（7）提供公开公平的对话机会，以解决组织冲突。

（8）鼓励员工互相监督，打击工作中夹杂的不良组织政治行为。

第四节　组织冲突

一、组织冲突的概念、种类及后果

组织冲突（organizational conflict）也是影响组织气氛和士气的重要因素之一。它是组织中员工之间、部门之间或员工与部门之间由于意见分歧或利益不一致而产生的心理和行为对抗，是一个从知觉到情绪，再到行为的心理演变过程。在这个过程中，个体或团队在各种条件的影响下首先意识到他或他们的利益正在受到另一个个体或团队的影响，产生知觉层面的冲突感受。如果情况并非知觉或想象的那样严重，或者对方的所作所为并没有对自己造成既成的后果，冲突就结束在知觉层面上。否则，就会发展演变为情绪上的对抗，乃至行为上的对立。冲突常常具有扩大化效应，一方面由于社会情境或他人的助长作用，冲突常常会不断升级；另一方面当事人双方的冲突行为会进一步影响双方的知觉和情绪，加深误解和情绪上的对抗，从而陷于恶性循环。因此，防微杜渐，尽快尽早地使冲突双方澄清误会、解决问题，是十分重要的。

那么，组织冲突会带来哪些后果呢？要回答这个问题，首先需要区分不同类型的组织冲突。根据冲突产生的基础，可以将其区分为任务指向性冲突（task-related conflict）和社会情绪性冲突（socioemotional conflict）。任务指向性冲突是由于在工作问题上的分歧所产生的冲突，是围绕工作任务而展开的，目的在于更好地完成任务，较少涉及个人评价、人际关系等敏感的社会情绪性问题。一般地说，这种冲突不仅是无害的，而且是有益的和必要的。它常常起到“催化剂”的作用，促使冲突双方乃至旁观者重新审视问题，作出更好的决策，发现完成任务的新方法、新途径，还可以增强组织的活力，活跃组织气氛，因此是一种富有建设性的冲突。正如有人所说的那样：“如果两个人是完全一致的，其中的一个就是多余的。”然而，社会情绪性冲突却常常具有强烈的破坏性。它是由于人际问题而引起的对人不对事的冲突，常常基于个人偏见或成见，并伴有浓烈的社会情绪色彩，目的是贬低对方而抬高自己。对组织而言，这种冲突会助长组织政治行为，分散冲突双方和其他组织成员的注意力，造成或加剧组织的紧张气氛，影响组织的正常业务和工作效率。对个人而言，这种冲突会给当事人双方带来巨大的心理压力，使其处于高度紧张的应激状态，降低其工作满意度，严重的还会产生工作倦怠、旷工、离职等心理和行为后果。

二、组织冲突的来源

在组织情境中，下列因素可能是冲突的起因或来源。

1．目标不相容

当组织中不同个人或部门的目标互不兼容、相互干扰时，就会引起冲突。目标对各自来说越是重要（特别是能带来经济奖赏时），引起的冲突就会越大。

2．差异化

当组织中的个体由于各自独特的背景和经历而持有不同的信念和态度时，差异化就产生了。差异化是冲突的重要来源。不同的教育背景、价值观念以及文化差异等都是导致冲突的重要因素。

3．任务的依赖性

所谓任务的依赖性，是指为实现目标团队成员需共享资源、发生互动的程度。在一些情况下，成员的最终所得还会取决于他们共同努力的绩效。通常，任务依赖性越强，发生冲突的几率就越高。

如图 17.4 所示，根据任务依赖性由低到高的不同，可以将不同团队之间的关系区分为三种类型：（1）共用型，指各工作团队需共用某些公共资源，但其他操作均可独立进行；（2）顺序型，指某一工作团队的输出会直接成为另一工作团队的输入；（3）交互型，指工作团队之间需要不断进行协作和交流。

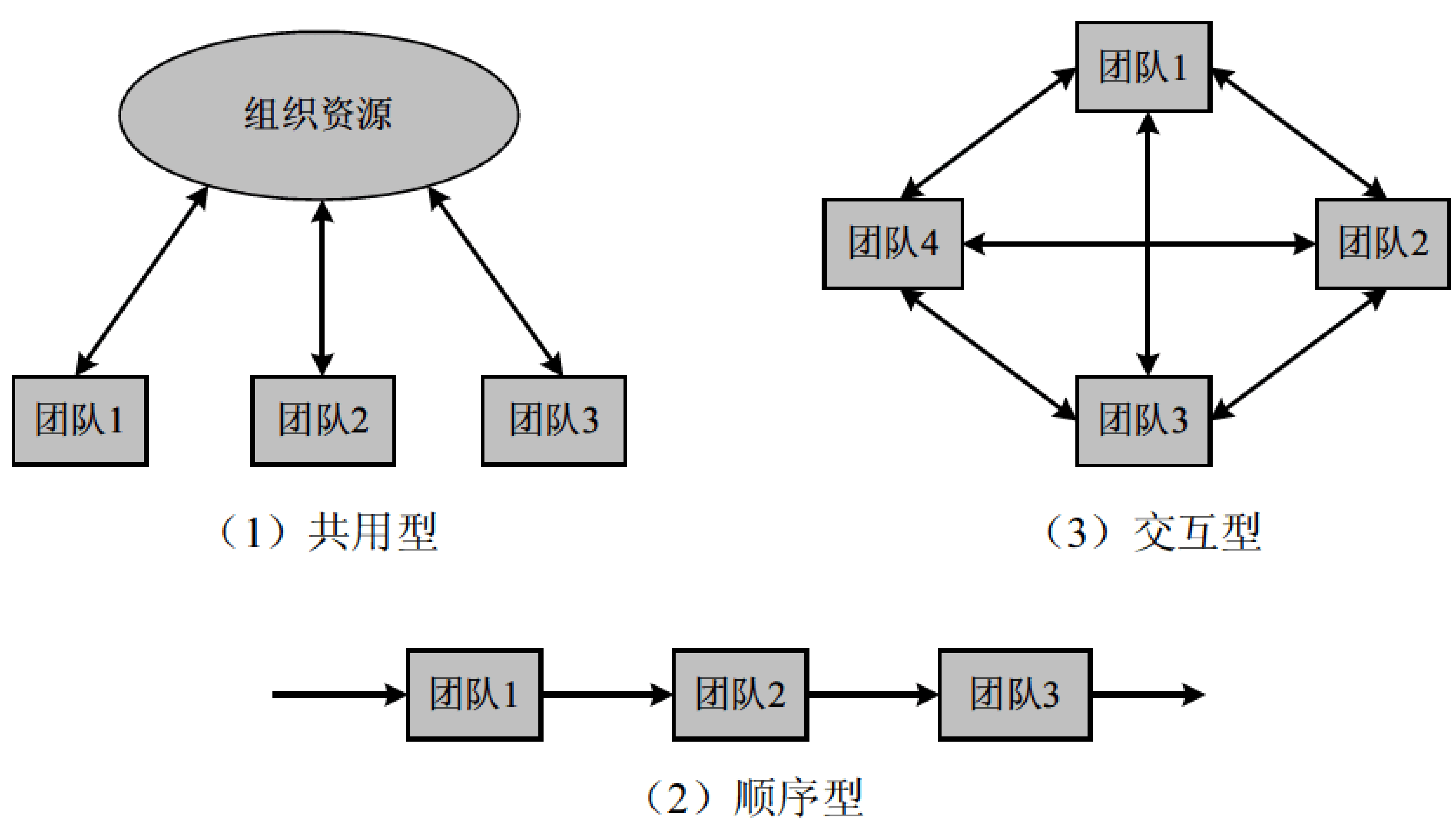


图 17.4　任务依赖性不同的团队关系（引自 Steven L. McShane & Mary Ann Von Glinow,2000）

4．资源匮乏

当资源缺乏，不足以满足所有工作部门的需要时，容易引发冲突。资源匮乏会使人们通过和那些也需要这些资源的人竞争来获得资源。

5．模糊性

模糊性是指工作进程中的不确定性，它会渲染团队之间相互竞争的气氛，增加目标相互干扰的可能性。若组织中有双方一致允诺和赞同的规则，则可以有效地控制由模糊性造成的潜在冲突。

6．沟通不当

人们缺乏有效沟通的机会、能力或动机，是导致冲突的重要原因。（1）若双方缺少沟通机会，则各自容易根据已有的心理定势来解释和预期对方的行为，而心理定势是一种先入之见，带有主观性，并伴有一定的情绪色彩，因此容易造成误解，引起冲突。此外，缺乏直接接触和交流机还会导致双方难以形成心理上的认同和移情。（2）缺乏沟通的技巧和能力也是引发冲突的重要因素。当一方以过激的方式表达不同意见时，对方很可能会同样报以不合作态度。如此一来，怨怨相报、恶性循环，会导致冲突升级。（3）交流上的不顺畅会削弱双方做进一步沟通的动机。社会情绪性冲突所带来的强烈的负面情绪使人们回避沟通，而更少的交流则会进一步强化已有的心理定势、加剧冲突。

三、冲突管理

所谓冲突管理（conflict management），是指采用一定的干预手段改变冲突的水平和形式，以最大限度地发挥其益处而抑制其害处。在组织情境中，通常可以从以下几个方面开展或加强冲突管理。

（一）确定适当的冲突管理风格

在人际互动过程中，组织成员对冲突的知觉、预期以及信念是各不相同的，因此首先应该有针对性地选择适当的冲突管理风格。

1．两种冲突管理取向

采用不同的立场会影响成员解决冲突的方式和风格。双赢取向（win-win orientation）认为双方最终能够寻找到一种互惠、共赢的方式来解决分歧；而输赢取向（win-lose orientation）则认为双方的共有资源是有限的，若一方赢得较多，另一方的利益就会相应地受损。在组织沟通过程中，若双方均持有输赢取向的立场，则容易激化冲突。事实上，在很多情况下，如果双方不以非此即彼的方式来看待冲突，而是以建设性的态度来理解分歧，则会有更多机会实现互惠和双赢。

2．五种冲突管理风格

根据组织成员介入冲突的方式，研究者区分了五种冲突管理风格。如图 17.5 所示，每种风格均可用两个维度上的不同水平组合来标识。维度一为武断性（assertiveness），代表成员试图满足自身利益的动机；维度二为合作性（cooperativeness），代表成员试图满足他人利益的动机。这两个维度形成了五种典型的冲突管理风格：

（1）协同（collaboration）：是指双方通过积极地解决问题来寻求互惠和共赢。其特征是双方乐于分享信息，并善于在此基础上发现共同点，找到最佳解决方法。通常，协同是

首选的冲突管理方式。但只有当双方没有完全对立的利益，且彼此有足够的信任和开放程度来分享信息时，协同才能有效地发挥作用。

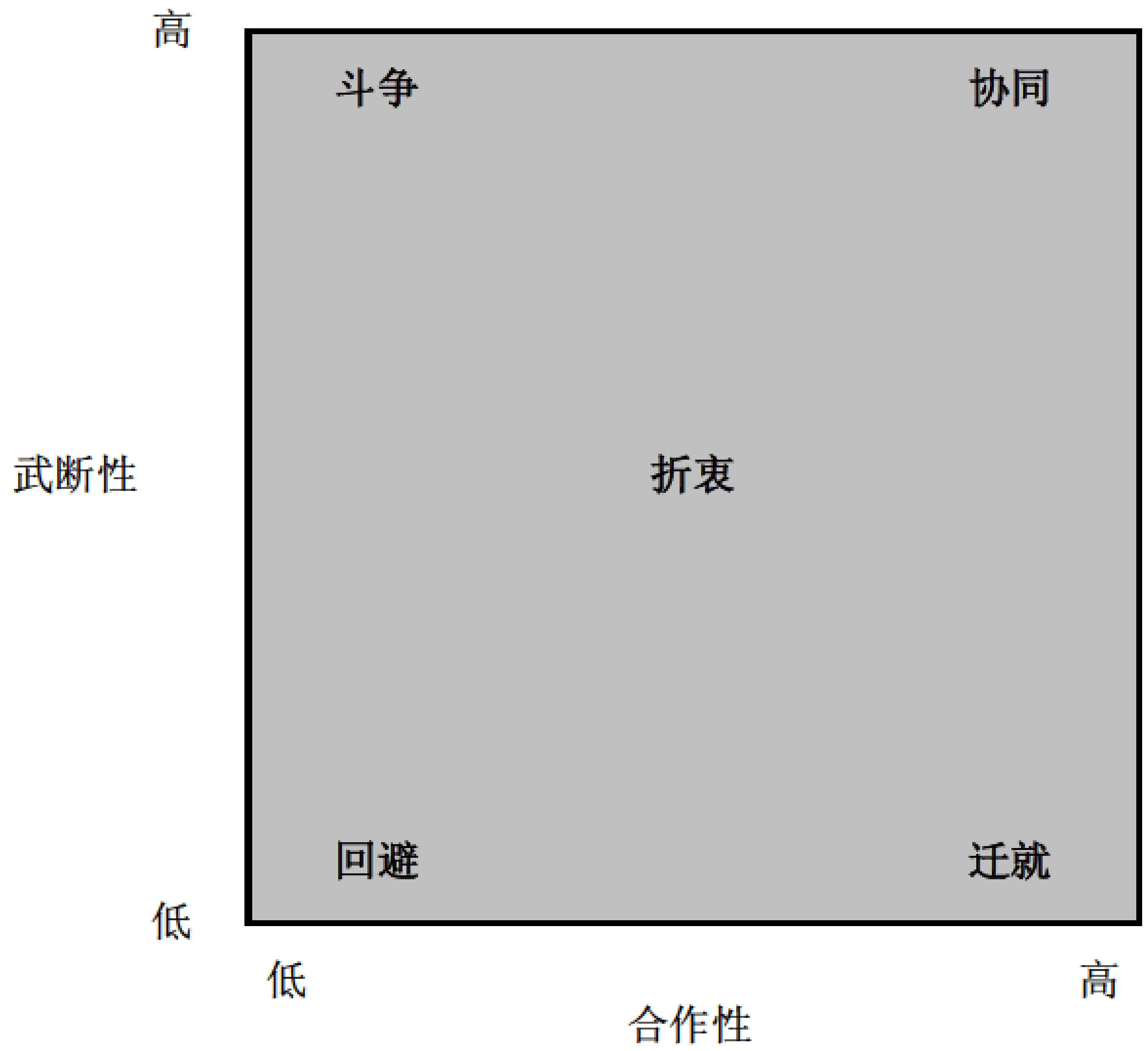


图 17.5　冲突管理风格（引自 Steven L. McShane & Mary Ann Von Glinow,2000）

（2）回避（avoidance）：是指试图通过逃避问题情境的方式来平息冲突。这种比较消极的冲突管理方式在应对不太紧要的问题时比较有效，此外，当问题需要冷处理时亦可采用回避作为权宜之计，以防止冲突进一步激化。但是回避无法从根本上解决问题，且容易导致自己和对方产生挫败感。

（3）斗争（competition）：是指以他人的利益为代价，试图在冲突中占上风。这种极端不合作的冲突管理方式通常并不是最佳解决方案。但是，当确信自己是正确的，且分歧需要在较短时间内解决时，斗争是必要的。

（4）迁就（accommodation）：是指完全屈从于他人的愿望，而忽视自身的利益。当对方权力相当强大或问题对于自身并不是太重要时，迁就是比较有效的方式。但它容易令对方得寸进尺，从长远看，迁就并不利于冲突的解决。

（5）折衷（compromise）：是指试图寻求一个中间位置，使自身的利益得失相当。这种方法比较适合难以共赢的情境。当双方势均力敌，且解决分歧的时间期限比较紧迫时，折衷是比较有效的。但是由于忽略了双方的共同利益，因此折衷往往难以产生非常令人满意的问题解决方法。

可以看出，协同是唯一完全体现双赢取向的风格。没有一种风格适用于所有的情境，而在每一种情境下，都有最适宜的冲突管理方式。因此，针对不同的情境采用不同的冲突管理风格，是冲突管理的精髓所在。

（二）选择合适的冲突管理策略

1．强调高级目标（superordinate goal）

高级目标是指超越冲突双方各自具体目标的更高一级的目标，是冲突双方服务和追求的共同目标。通过各种方法突出高级目标的重要性，有利于增强组织凝聚力，减少社会情绪性冲突。在解决由目标不兼容和差异化造成的冲突时，此种策略的作用尤为显著。通过提高成员对组织共同目标的忠诚度，可以有效地解决由部门目标不一致造成的分歧；在异质性团队中，若成员理解并认同了组织的共同目标，则能够有效地避免差异带来的潜在冲突，使团队成员能够各施所能，全力为组织的共同目标服务。但由于该策略仅仅是通过引入一个参照目标来抵制差异化，因此它无法从根本上消除组织内部各种潜在的多样性及其负面影响。

2．减少差异化

减少差异化是指通过改变或消除导致差异的各种条件，直接抵制分化。它包括消除形式上的差别（如统一工作制服等）和培养共同经历（如鼓励通才式职业定向、实施团队轮换制）等方法。

3．增进沟通和理解

有效的沟通对冲突管理是至关重要的，它能消除刻板印象带来的偏见和负面情绪，增进彼此的理性认识。在组织管理中，常用的沟通方法有对话法（dialogue）和组间镜像法（intergroup mirroring）。对话法是指通过团队成员之间正式或非正式的交谈来讨论彼此的分歧，在了解各自基本设想的基础上建构团队共同的思维模式。组间镜像法一般适用于双方冲突已恶化到公开对立地步的情形，通常需要管理者有计划、有步骤地进行干预。其目标旨在为冲突各方提供一个充分表达各自观点、讨论分歧的机会，并最终通过改变错误观念来找到改善双方关系的途径。

4．降低任务依赖性

降低任务依赖性可以有效减少冲突发生的几率。对于共用型任务依赖，可以采用分离共用资源的方法；而对于顺序型和交互型任务依赖，则可以采用合并任务的方式来降低任务依赖性。此外，还可以通过建立缓冲带的方法（如建立专门的调解委员会）来协调不同部门的工作。

5．增加资源

解决由资源匮乏导致的冲突时，增加资源无疑是最直接、最有效的方法。当然管理者需权衡增加资源的成本及冲突带来的损失。

6．明确规则与程序

明确规则与程序能够有效解决由模糊性带来的冲突，尤其是当资源匮乏时，如何分配和利用资源需要作出明确的规定。这有利于消除误解，建立公平、公正的工作环境，增强组织的凝聚力。

（三）采取必要的冲突管理措施

1．通过谈判解决冲突

谈判是指冲突各方试图通过重新界定他们之间相互依赖关系的条件来解决目标分歧。只要人们之间存在着彼此依赖的关系，谈判就是可能的，也是解决冲突的有效方法。

有研究者通过图 17.6 所示的交易区模型（bargaining zone model）来说明和解释谈判行为。在谈判过程中，冲突双方通常会确立三个基本点：起始点、目标点和阻抗点。起始点是各方开始时的报价或要求，代表了各自理想的目标或状态；目标点代表各方比较现实的目标；而阻抗点则代表谈判各方的底线。谈判之初，双方分别陈述自己的起始要求，随着谈判的进行，各方都会有所让步，要求也会相应地发生变化。只有当双方的要求均落入中间的交易区时，谈判才会达成协议，冲突才能得以解决。

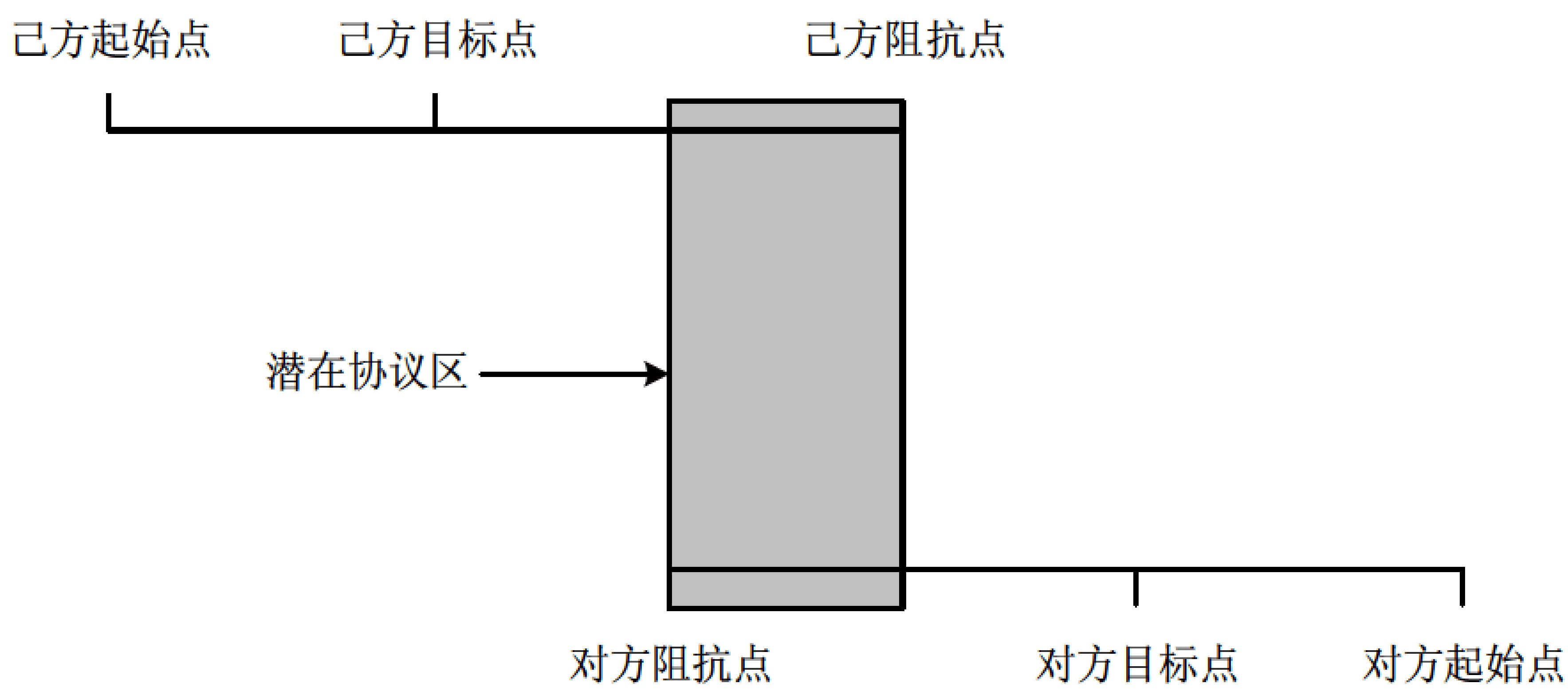


图 17.6　谈判行为的交易区模型（引自 Steven L. McShane & Mary Ann Von Glinow,2000）

以下两类因素会影响谈判的效果和质量：

（1）谈判的情境因素

①谈判地点：在熟悉的环境下进行“主场”谈判可以使谈判者从容不迫，无需应付旅途劳顿及陌生环境带来的不便。鉴于此，很多谈判者会预先协商在某个中立地进行谈判。随着计算机技术的发展，双方可以方便地实现远程交流，这时不再存在谈判地点的问题。但是由于远程谈判无法面对面交流，很多微妙的非言语线索容易被忽略，信息传达易被曲解，加之谈判问题本身的复杂性和不确定性，因此重要的谈判通常不采用这种远程交流方式。

②谈判场所的物理布局：谈判双方的距离、座位安排等布局形式会影响谈判者的态度和取向。通常，面对面的座次安排会加重对抗的意味。

③时间进程和最后期限：谈判时间越久，双方就会越希望能够解决分歧、达成协议，但同时也容易出现无原则让步的情况。最后期限会给谈判双方带来时间压力，通常随着最后期限的逼近，双方更容易做出让步，因此好的谈判者会善于运用最后期限来谋求利益的最大化。

④观众特点：当与谈判结果有利害关系的群体在场时，谈判者通常会表现得更加强硬，更不愿意做出让步。

（2）谈判者的行为因素

①计划与目标设定：研究发现，合理的计划与目标设定能够提高谈判质量。谈判者预先应仔细考虑己方起始点、目标点和阻抗点，并要统一团队内部的意见。

②收集信息：在试图被对方理解之前，先要努力理解对方。在谈判过程中，应通过仔细聆听和积极询问来收集对方的信息，只有了解了对方的需要和利益，才能选择适当的策略。

③有效交流：谈判者的交流方式应能有效地维系彼此关系，有效的交流方式如将焦点集中于事务本身，避免过激言行，采用有说服力的表达方式等。

④适度让步：适度让步有利于双方达成协议，让步的程度取决于对方的期望和彼此信任的程度。若过分强硬，则无法显示诚意，会破坏双方关系；而过分让步则会令对方得寸进尺。

2．第三方介入

第三方介入是指相对中立的第三方作为调解人或仲裁者，帮助冲突各方解决分歧。第三方介入主要应该达到以下几个目标：

（1）效率高，即以最少的组织资源来迅速解决争端。

（2）效果好，即找到从长远看能够解决冲突根源的最佳方法。

（3）结果公平，即冲突各方均认为第三方介入提供的解决方法是公平的。

（4）程序公正，即冲突各方均认为解决分歧的过程是公正的。

根据第三方对冲突解决过程和结果的控制程度，可以将第三方介入分为三类：

（1）调解（mediation）：调解者对冲突解决过程拥有较高水平的控制，而对于冲突解决的最终方案，控制性较低。在调解过程中，第三方积极管理冲突各方互动的过程和背景，但最终解决方案仍需冲突双方做出决定。

（2）仲裁（arbitration）：仲裁者对冲突解决的最终方案拥有较高水平的控制，而对于冲突解决过程，控制性较低。仲裁者会对冲突各方做出强制执行的决定，而冲突解决过程则主要按照既定程序和规则进行。

（3）审判（inquisition）：审判者对冲突解决的过程和最终方案均拥有较高水平的控制。在审判过程中，第三方不但决定最终的冲突解决方案，而且决定冲突解决的过程将如何进行。

在组织管理中，当管理者充当第三方介入到部门或个人冲突中时，需根据不同情境扮演不同角色，才能有效地帮助冲突各方化解分歧。研究发现，若管理者过多地充当审判者角色，往往不利于冲突的解决。因为在这种情况下，管理者搜集的信息往往有限，因此判决的有效性会大打折扣，而且由于组织成员对整个过程和决定缺乏控制，因此他们会认为审判有失公平。与此相比，调解往往会使组织成员产生最大的满意度，但有时其效率不佳。仲裁比较适用于优先考虑组织目标而不是个体目标的情况，且由于其程序公正，容易被组织成员所接受。

本章小结：

1．影响组织气氛和士气的因素主要包括组织文化、组织权力、组织政治、组织冲突。

2．组织文化是指支配组织成员在面对问题和机遇时进行思考和行动的共同设想、价值观及信念的基本模式。通过表征组织文化的具体符号和标志可以间接地理解组织文化的内涵。除了被绝大多数组织成员接受的主流文化外，组织中还存在着相对独立的各种亚文化（包括与主流文化所倡导的核心价值观不一致的反文化）。

3．组织文化具有导向、约束、凝聚、激励和辐射等功能。影响组织文化功能发挥的因素有文化强度、文化匹配性及文化适应性。

4．在组织合并或重组过程中，应预先考察两个组织文化上的兼容性。实现组织文化融合的策略有同化、去文化、整合和分离等。

5．巩固和发展组织文化的途径主要有：通过创始人与继任领导者的行为活动、推行与组织文化一致的奖赏体制、维持稳定的员工人数、建立有效的文化传播网络、挑选员工并对其进行组织社会化。

6．组织权力是指个体、团队或组织所具备的一种影响他人的潜在能量。权力产生的最基本的前提是一方或双方意识到对方的存在及其能量。组织权力的来源主要有法定权力、奖赏权力、强制权力、专家权力和参照权力。此外，有研究者提出了信息权力的概念。组织权力的可替代性、中心性、自主性和可见性影响其作用的发挥。

7．组织政治是指通过一定的自主行为来影响他人以达到自己的目的。多数情况下，组织政治无益于组织的发展，但并非所有的组织政治行为都有损组织发展。对组织政治行为进行价值判断应考虑三个原则：实用原则、权利保障原则和公平分配原则。

8．常见的组织政治行为包括攻击或责备他人、控制信息、形成联盟或帮派、培植网络、施恩图报及印象管理。组织环境和个体特征都会影响组织政治行为。应对不良组织政治行为的措施包括：尽可能保证充足的关键资源、制定资源获得和利用的明确而具体的规则、建立畅通无阻的信息交流系统、加强与员工的沟通、必要时重构团队和组织规范、避免权谋者当政、提供公开公平的对话机会、鼓励员工互相监督等。

9．组织冲突是组织中员工之间、部门之间或员工与部门之间由于意见分歧或利益不一致而产生的心理和行为对抗，是一个从知觉到情绪，再到行为的心理演变过程。其来源主要有目标不兼容、差异化、任务依赖性、资源匮乏、模糊性和沟通不当。

10．管理组织冲突可以从确定适当的冲突管理风格、选择合适的冲突管理策略及采取必要的冲突管理措施三方面入手。协同、回避、斗争、迁就、折衷是常见的冲突管理风格；强调高级目标、减少差异化、增进沟通和理解、降低任务依赖性、增加资源、明确规则与程序是常用的冲突管理策略；谈判和第三方介入是可以采取的冲突管理措施。

本章思考题：

1. 简析组织文化的概念、功能。如何评判组织文化的优劣？
2. 试比较实现和促进组织文化融合的四种策略。举例说明其中的一种。
3. 试述保持和巩固组织文化的方法和途径。
4. 分析组织权力的概念及其发生作用的机制。
5. 简析组织政治行为的利弊及其价值判断原则。
6. 常见的组织政治行为有哪些？举例说明如何管理组织政治行为。
7. 简析组织冲突的概念、种类及利弊。
8. 组织冲突的来源有哪些？举例说明如何管理组织冲突。

第十八章　组织变革和发展

面对当今快速发展和变化的环境，组织必须随时调整战略、方针、结构、政策，以增强自身的生存和竞争能力。本章将对组织变革和发展的相关问题加以介绍和分析。

第一节　组织变革

一、组织变革的外部动因

回顾组织的发展历史，每一种组织形式和管理方式都与当时的政治、经济、社会、文化和技术等因素相适应，都有其合理性。传统的企业组织所采用的直线制、职能制、直线职能制、事业部制、分权制和矩阵制等形式，适应于资本主义工业化大生产时代的经济社会环境，极大地促进了经济和社会的发展。但是，随着知识经济时代的到来，企业所面临的各种环境条件发生了深刻的变化，最突出的就是“三化”和“三 C”。“三化”是指全球化、市场化和信息化；“三 C”是指 change、competition 和 customer，即日新月异的变化、白热化的竞争和以客户为中心。在这种形势下，传统的企业组织管理模式已经无法适应新的要求，因此组织变革成为必然的、唯一的选择。具体而言，以下几个方面的变化构成了组织变革的外部动因。

（一）生产技术的发展

随着世界范围内生产技术的突飞猛进，许多商品和服务的生产成本在逐渐降低，质量和产量却在迅速提高和增加。商品的产量远远大于消费者的购买需求，使得消费者在购买商品和服务时拥有了充分的选择权，从而导致由卖方市场向买方市场的转变，真正的客户经济时代已经到来。企业只有不断调整、变革战略和结构，才能与不断变化的市场需求保持一致，吸引并保留有价值的客户。

在技术高度发达的今天，生产技术不断复杂化，分工也越来越精细。许多管理者发现他们对技术的了解越来越有限，而最了解生产技术和服务技巧的是处于生产一线的员工。正因为如此，现代的企业组织不得不由以前的集权制向授权制转变。基层的员工获得了相应的权力，从而使得生产和服务更加快速和周到。另一方面，技术的复杂化和快速发展使得企业单独行动的风险大大增加。为了降低风险，竞争对手之间的合作成为必要，联合开发和研制的管理模式越来越多地被不同企业所采用，战略联盟、企业集群等新的组织形式也登上了历史舞台。

（二）信息技术的进步

以互联网为核心的信息技术的发展对企业内、外部环境的影响是深远的。

首先，信息技术的进步使组织结构变革成为必需。企业中信息的分布对企业中的权力分配有着重要的影响。在传统的组织模式中，企业层级非常多，最高决策者依赖于中层管理者将企业的信息自下而上传递上来，据此做出企业经营管理决策；同时中层管理者又将企业高层的决策自上而下传递，并监督实施。在信息技术高度发达的今天，这种信息传递的质量和速度都不能满足企业竞争和发展的需要。在此种情况下，变革组织结构就显得极为重要。信息技术的引入，使得企业的信息无需通过中间管理层就可以快速高效地传递到企业的决策层。中层管理者减少了，企业组织的层级得到压缩，组织结构越来越扁平化了。

其次，信息技术的进步使组织形式的变革成为必需。在传统的企业组织形式中，信息的典型传递途径是链型、Y 型和轮型等形式，这也就决定了员工的工作也应该按照相应的方式来组织。但是，在信息技术高度发达的今天，信息在组织内部的传递方式已经成为全通道型，即由于工作复杂性的加剧，组织内横向的信息交流非常频繁，因为只有这样，组织才能适应不断变化的市场环境，得以生存下去。在这种情况下，企业的工作形式不得不发生变革，由跨越职能团队的形式来组织。

第三，信息技术的进步使顾客服务策略的变革成为必需。随着信息时代的到来，顾客的消费模式发生了巨大变化，一方面，他们可以利用先进的信息技术搜寻自己中意的产品，通过比较做出理性的选择；另一方面，消费者被大量的媒体所包围，市场上出现的问题被各类媒体快速报道，各种信息互动的结果使得消费者之间的影响也在不断增强。在这种情况下，生产商和供应商面临越来越大的压力，开始密切关注客户和竞争对手，过去以生产为中心的组织管理形式已经不再适合企业发展，以客户为中心的组织管理形式成为必然的选择。这种组织管理形式要求企业内部打破部门之间的界限，各种不同技能的员工相互之间充分地进行交流，从而从不同的角度认识企业环境，丰富企业知识。

（三）顾客需求的变化

知识经济时代，顾客的需求呈现出个性化和高级化的趋势，而消费者的需求又是企业生产和服务的起点和归宿。所以，企业再也不能关起门来，按照自己固有的方式运转，而应充分重视和利用消费者的个性化需求，将其纳入到企业的价值生产链中。越来越多的企业意识到应该以市场为导向，从多层次和多渠道建立与市场的联系，从而导致企业的组织结构不断分散化。同时，为了应对消费层次的高级化，企业必须不断拓展自身的知识创新能力。这就要求企业在进行组织设计时，充分考虑激发员工的创造性，鼓励员工进行自我监督和自我管理，同时还要管理好企业内部共享的知识，方便员工之间进行交流，从而提高企业的整体创新能力。

另一方面，作为企业内部顾客的员工的需求也在不断发生变化，这也是企业所必须重视的。雇员是最终外部顾客价值的创造者，企业在为客户创造卓越价值的同时，还必须为雇员创造价值。这也要求企业在进行组织设计时，充分考虑实现雇员价值方面的要求，如采用个性化的激励计划、充分授权、建立自主工作团队等。

（四）价值创造的新趋势

在传统的组织管理中，业务操作过程被分割成互不关联的片断，每个片断都处在相互隔离的部门之中。在这种情况下，组织管理流程中存在着相当数量没有产生价值的无效行动，难以实现以价值为核心的管理模式。为了更好地向顾客传递价值，企业必须打破内部的限制和障碍，围绕业务流程，将企业的硬件设施、奖励制度、组织结构进行整合。目前，不少世界著名企业，如惠普、百事可乐、克莱斯勒、福特、壳牌石油等，都不同程度地实施了企业再造计划，取得了令人瞩目的成效。

创造价值的要求还产生了虚拟整合。这意味着企业之间的“墙”被推倒了，不同企业之间紧密合作，各自完成自己最具优势、成本最低的工作，最终为客户创造最大的价值。之所以如此，是因为每个企业都不可能精通所有的工作。那些恪守垂直整合的企业必定会因为承担了自己不擅长的工作，不能充分发挥自己的优势，从而导致成本上升。

二、组织变革过程

（一）从勒温的力场分析模型看组织变革过程

1．组织变革的驱动力与抵抗力

按照图 18.1 所示的勒温（Lewin）的力场分析模型，任何一个组织都是一个“场”，其中交织着各种力量的斗争。当各种力量势均力敌时，组织就处于一种平衡或稳定状态，呈现出特定的组织结构型态。所谓组织变革，就是组织由目前状态向预期状态的转变过程。在这个过程中，促使组织变革的驱动力与抵制组织变革的抵抗力在不同的水平上展开斗争，在不同的层次上取得平衡，逐渐向预期状态过渡，直到在预期状态上取得平衡。

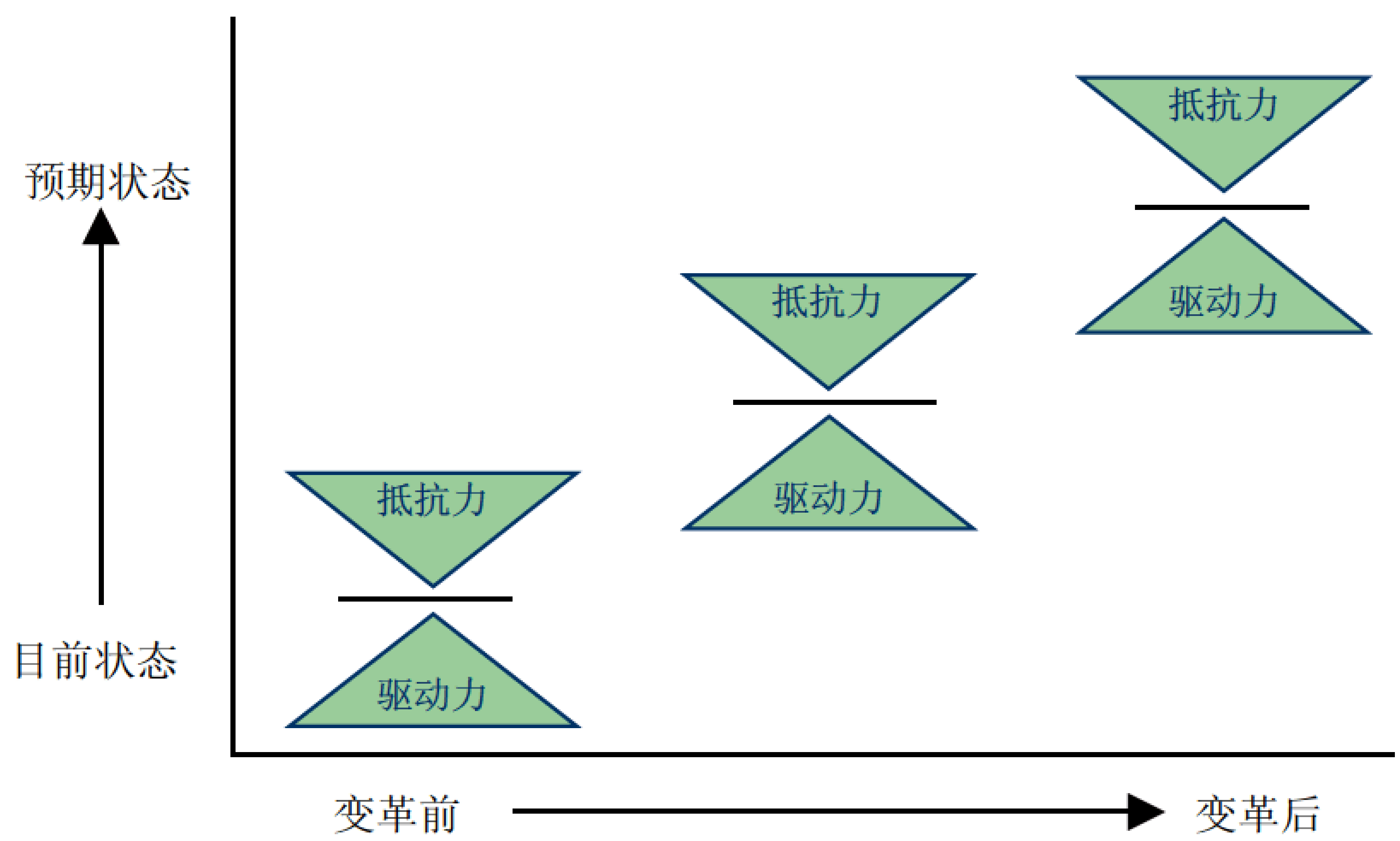


图 18.1　勒温的力场分析模型（引自 Steven L. McShane & Mary Ann Von Glinow,2000）

组织变革的驱动力包括组织外部的驱动力和组织内部的驱动力两类。前文所述的组织

变革的外部动因都属于组织变革的外部驱动力，除了前述四个方面的外部动因之外，组织变革的外部驱动力还可能包括上级组织的指示或要求等；组织内部的驱动力包括组织内的权力和政治斗争、组织冲突、部门竞争、领导意志等。

组织变革的抵抗力既可能来自组织外部，又可能来自组织内部。组织外部的抵抗力包括上级组织的阻力、竞争对手的阻挠、客户的抱怨和不满等；组织内部的抵抗力主要来自员工的担心、忧虑所引起的消极或抵制行为。

研究表明，组织变革过程中，员工通常会由于以下的担心和忧虑而产生不配合甚或抵制行为：（1）直接代价：担心失去既得利益和现有地位；（2）爱面子：证明自己正确和怕丢面子的倾向；（3）规避不确定性和风险：担心不能适应变化而遭淘汰；（4）打破了惯例：不愿放弃不再适当的行为惯例，学习新的角色模式；（5）不一致的组织系统：担心新的奖励系统、选拔系统、培训系统给自己带来麻烦；（6）不一致的团队动力：担心不能适应新的团队规范和人际关系。

2．解冻、变革和重新冻结

（1）解冻

即引入变革动因，打破现有的组织平衡或稳定状态。解冻是变革的前奏，是推行变革措施前的酝酿、宣传、说服和教育活动，其目的是统一思想，扫除变革的障碍。在这个过程中，组织一方面要设法增强变革的驱动力，另一方面要设法降低变革的抵抗力。

增强变革的驱动力，关键是要让员工认识到变革的迫切性。一般地说，可以从以下几方面入手来增强变革的驱动力：①让员工理解竞争对手的情况；②让员工了解顾客需求的变化；③让员工理解政府的政策。

降低变革的抵抗力，即针对员工的担心、忧虑和消极抵制情绪展开一系列活动，通常包括以下一些管理措施：①通过沟通和对话消除员工的顾虑；②通过培训改变员工的态度和提高员工适应变化的能力；③让员工参与变革的决策过程；④通过应激管理缓解员工的压力和紧张情绪；⑤通过谈判解决棘手的问题；⑥如果时间紧迫，机会成本过高，对少数经过上述工作仍固执己见的员工采取强制措施。

（2）变革

即按计划实施和推行变革措施，过渡到预期状态。从形式上看，变革可能是渐进的、微妙的变化过程，也可能是突进的、戏剧性的变化过程。无论采取哪种形式，员工都必须学习和内化新的行为模式。

（3）重新冻结

即巩固变革成果，防止回到以前的角色和行为模式。为了做到这一点，管理上应该注意以下几点：①组织结构要有助于锚定员工的角色和行为模式；②奖励系统要鼓励所期许的行为；③信息（宣传）系统要表扬和赞许新的行为；④反馈系统要帮助员工知道自己做得如何、还应怎么做。

（二）组织实施和推行变革的重要举措

1．制定战略规划：变革促进者应该事先制定清晰的、明确的未来状态的蓝图，明确角色知觉，指引未来行为。

2．确定变革代理人：变革代理人，即有足够知识和权力指导和促进变革的人。其作用是斡旋、联络、沟通，促成变革，对未来状态作出承诺。人选可以是专家、顾问、组织领导、员工等。

3．推广变革成果：就是用试验性项目测试变革过程，然后将从试验中得到的经验推广到组织的其他部门。其益处在于具有更大的弹性和更小的风险。试验中得到的经验能否推广取决于一些条件，包括：试验项目要在一两年内获得成功，有明显效果；相关措施不能太模糊，也不能太精确；试验组无离职者等。

三、组织变革的结果

（一）从组织结构看组织变革结果

大多数组织变革理论都是从组织结构的变革推演而来的，它们都从宏观角度出发，认为组织结构的变化是组织变革的关键所在。近年来，通过组织变革出现的新的组织结构形式主要有以下几种：

1．柔性动态型组织

为了适应环境的变化，现代企业迫切需要创建柔性组织系统，以平衡集权与分权、自主权与控制权、主管与参谋的关系，增强对意外风险的不断适应和调整能力。具体形式有：

（1）弹性动态型组织结构。传统的组织结构长期处于静止状态，群体由固定人员组成，个体永久性固定在一个岗位上，组织缺乏活力，员工的潜能被禁锢。这种机械性组织结构若处在较为稳定的外界环境中，尚可勉强应付，但若处在变化较快的环境中，则会出现反应缓慢、组织创造力衰退的情况。

近些年来，不少企业为了适应环境的发展和变化，已经开始采用有机性组织结构，如弹性动态型组织结构。这种组织结构不再按照专业设置科室，而是根据任务的需要和变化来设置科室。除办公室、人力资源部等常设机构外，其他机构依据任务的变化随时调整、整合和重构，企业中不同科室、不同岗位、不同权力、不同机构依据不同情况而处在动态过程之中。其优点是：为组织机构增添了活力，增强了开放性；员工能够交流经验、方法和信息；工作有新鲜感、挑战性；员工潜能得到开发，组织“进可攻，退可守”，灵活性大大增强。

（2）浮动升降型组织结构。浮动升降型组织结构中的“浮动”主要体现在两个方面：一是责任升级化。如韩国大宇集团，在公司内设置分公司，分公司实行独立核算，并将许多裁决权授予分公司。这提高了内设机构的责任级别，逐步从一般管理中心上升到成本中心，从成本中心上升到利润中心，从利润中心上升到投资中心，从而不断增强企业抗风险能力。二是级别浮动化。韩国有许多商社将公司内设机构划分为“部”和“事业部”两大类别。如果某“部”效益增加，就提升为“事业部”，其待遇和决策权相应上升；如果“事业部”效益下滑就下降为“部”，其决策权也随之下降。这种方法将竞争机制引入内设组织机构，以绩效好坏决定其机构级别及待遇的升降，使机构处于上下浮动状态。企业内设组

织浮动升降化，一方面使企业某些权力下放，有助于下级部门创造性地完成组织目标；另一方面组织机构引入了竞争机制，给员工施加压力，使他们更加勤奋努力。

（3）小型发散式组织结构。小型发散式组织结构，就是将人数众多、规模庞大的企业拆分开来，形成多家人数较少、规模较小的企业，这也是当今企业组织结构变革的一大趋向。其独特的功能在于：首先，体现了集权与分权的平衡。大权相对集中，从组织长远发展考虑，所有行动都需要密切配合，需要集中进行战略性决策等；但是小权又要相对分散，下级组织拥有主动权，有利于发挥下级组织及员工对产品、市场、地区比较熟悉的优势。其次，体现了较强的抗风险能力。大型企业分散化，可以发挥全面性、动态性和发散性等特点，各卫星企业在管理方法、方向、内容、结局上具有多样性。例如，三九集团下属 100 多家企业，形成了以药业为中心，农业、旅游业、酒业、食品加工业、房地产业、连锁商业、汽车制造业等八大产业并举的多元化发展格局。如果某一产业不景气，可以转产，不会从总体上影响集团公司的效益，船小好掉头，抗风险能力增强了。最后，这种组织结构有利于管理人员能力的提高。在各分公司，管理决策层职务轮换，可以培养全面的高级管理人才；中层管理者轮换可以培养复合型人才；基层管理者轮换可以提高其专业工作能力。当然，企业小型化也有不足之处，资金和权力分散，不能集中财力、物力、人力从事集团化经营。

2．结构异化型组织

在相当长的一段时间内，我国企业采用的是直线职能式组织结构形式，这在环境稳定性较强的条件下曾起过一定作用，但是这种组织结构形式具有容易产生专断作风，淡薄下级参与意识，部门之间协调意识较差，不能随环境的变动而变动等弱点。现代企业组织趋向于结构异化，其主要表现形式有：

（1）扁平长方型组织结构。扁平长方型组织结构的特征表现在：管理层次少，管理费用低，利于精兵简政；管理跨度较大，员工与员工之间平等关系占优势；组织内平行关系多，垂直关系少。这样的组织形式可以使决策权下移，上下级沟通更加顺畅，民意可直接通达于上层，下层也可直接了解上层领导的意图；可以使组织形成整体互动，创造巨大经济效益。组织外型扁平化已经受到中外企业的普遍青睐，GE 集团总公司从工厂车间到行政办公室总部的管理层已经从 9 层减为 4 层。日本 50%以上的企业组织运行程序愈来愈少，业务流速愈来愈快。

当然，扁平长方型组织结构也有其不足之处，如管理者工作量大，管理难度高，权力较分散，严格监督下级的程度较低等。

（2）倒金字塔型组织结构。随着生产技术的发展及员工素质的提高，目前的组织改变了上层决策、中层管理、基层执行的传统组织管理格局，形成了组织机构的倒金字塔化。这样的组织形式将现场决策权、管理权、考评权交给下级，可以激发下级的荣誉感、责任感，发掘下级的潜能，促使一线工作人员主动、积极、灵活地处理工作问题。由瑞典、挪威、丹麦三国共管的斯堪的纳维亚航空公司就形成了上层为战略目标制定和监督层，中层为管理层，基层为现场决策层的组织管理新格局。

倒金字塔型组织结构的实施是有一定条件的：从组织成员的构成上看，受过高等教育、

文化修养较高的白领员工应占多数；从企业性质上看，企业产品含金量较高的高科技企业更为适合。深圳华为集团公司的 6000 名员工中博士、硕士有 3600 多人，占 60%以上；地处北京中关村的美禾集团高学历员工占总人数的 90%以上；武汉的红桃 K 集团具有大学及大学以上学历者占总员工人数的 76.6%，这样的企业采用倒金字塔型组织结构可以使高素质人才创造出一般企业无法比拟的效益。

3．虚拟网络型组织

随着贸易经营范围的扩大、网络科技的发展，虚拟、寄生、网络型组织结构也日益成为组织变革的新趋向，主要表现为以下三种形式：

（1）委托代理型组织结构。委托代理型组织结构是在严格控制组织本身人员和机构数量的基础上，将部分非核心工作任务委托给其他组织代为完成的一种组织形式。我国云南玉溪烟厂除保留烟丝这个核心技术外，其他如烟卷纸、包装纸、过滤嘴等生产工艺与流程全部外包给乡镇企业；松下电器集团下属的先锋公司把影碟机委托给十家视听机器厂生产；耐克公司本身只有一个小工厂，只负责设计和销售。这种尽可能将核心产业和技术以外的部分外包的委托经营方式具有两个方面的优点：一方面，可以克服组织机构过于庞大、人员过多而造成的官僚化、成本过高等缺陷；另一方面，从战略高度上看，它符合“一、二、三”的管理格局（“一”是指企业要有一元化的核心领导中枢；“二”是指企业既要有核心产业、核心技术、核心管理，又要涉足其他经营领域；“三”是指企业要有保命区、形象区和风险区）。

（2）网络科技型组织结构。互联网的出现，使人们产生了“天涯若比邻”的新视野，“网络化”组织机构应运而生。这种组织机构包含两层含义：一方面，在企业内部，形成网络化联系。企业内部执行层的平行机构增加，每个平行机构通过电脑直接与决策层保持联系，形成企业内部的网络化组织结构。另一方面，在企业外部，形成了网络化指挥中枢。也就是说，企业通过发展连锁经营、代理商等方式形成庞大的销售网络。例如，美国有个网上花店，员工才 21 人，年销售额达 3 亿美元。日本花王公司 80%的产品是通过其庞大的销售网络卖出的。麦当劳已经在世界 100 多个国家和地区发展了连锁经营店。在 HP、IBM 等公司，老板随时可以通过电脑与远在天涯的雇员对话交谈。

总之，无论是企业内部还是外部网络化机构，既强化了指挥中枢，又给予执行层参与企业决策的机会，满足了员工的参与感，淡化了领导层与员工的界限，还为企业减少了开支以及人与人之间的内耗，提高了企业效益。

（3）虚拟联盟型组织结构。虚拟联盟型组织结构也是当前世界企业组织变革发展的新亮点。中国台湾地区的洪成公司创造了名牌休闲鞋，但没有自己的生产线；香港“鳄鱼”牌服装，其代理商只带设计图、品牌和质量检测标准三样东西，就可以与内地合作。这种没有自身生产设备和生产经营过程，依靠别人的力量来赚钱的经营形式称为虚拟经营，其组织机构称之为虚拟组织。虚拟组织多采用虚拟联盟型组织形式，包括战略联盟、知识联盟、技术联盟、产品联盟等。

虚拟联盟型组织结构有其不同于其他组织机构的特征、机制与功能，主要表现在：企业实体性组织与长期固定人员逐步减少，而临时性组织和临时性人员增加；组织决策的集

中化程度很高，而部门化程度比较低，甚至根本就不存在什么实体部门；组织之间可以通过核心优势的联盟而结合在一起；组织有较大的灵活性，可以利用与之联合的其他企业的优势，也可以转让本企业劣势部门；可以变一级法人市场为多级法人市场，变一个积极性为多个积极性，变一条销售渠道为多条销售渠道；可以达成信息、技术、产品、知识的交流和互补；有助于企业利用其他企业优势产品、技术，抓住机遇，降低成本，迅速收回投资，获取利润等。

（二）从个人特征看组织变革结果

对员工来说，组织变革是重要的应激事件，变革使员工处于高度的紧张和焦虑状态，他们必须在认知、情绪和行为上做出重大的调整，才能有效应付变革带来的变化。

个体对于组织变革的核心反应之一，就是他们在多大程度上能够适应变革给自己工作和生活带来的不确定性。研究表明，当个体采取以问题为中心的应付策略，即直接应对应激源时，其心理操作和外显行为更为有效；相反，如果采用以情绪为中心的应付策略，即着眼于应激源所引发的情绪变化，则会引起不良的反应。换句话说，将注意力放在解决问题、改变自身以适应组织变革的个体，更有可能在组织变革中获得成功；而过分关注自己的不适感、失落感、不满意感等负面情绪的个体，则有可能被组织所淹没甚或淘汰。研究发现，影响组织成员应对策略选择的是个体自身的认知图式或知识结构，而个人的认知图式又深受其人格特质的影响。人格特质、认知图式、应付方式和对变革适应结果之间的关系可以用图 18.2 来表示。

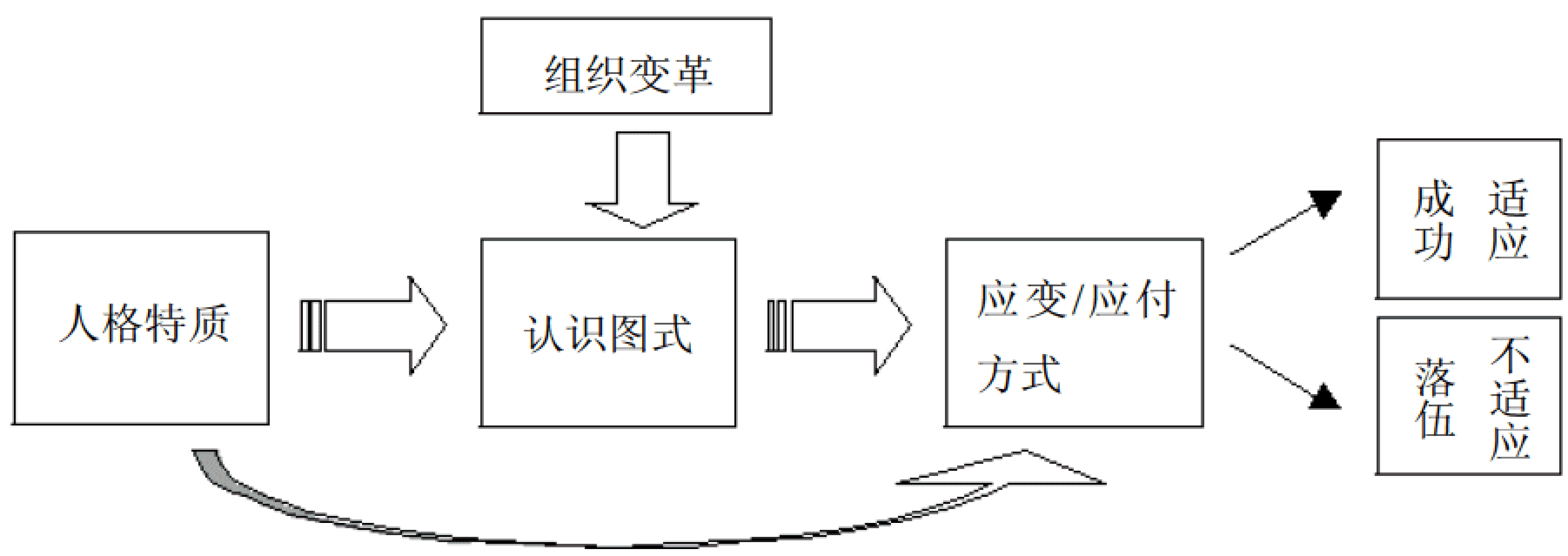


图 18.2　组织变革中个体的人格变量与应对方式之间的关系（引自王黎、张建新，2000）

与组织变革关系密切的主要人格特质有以下六种。

（1）控制源。所谓控制源，是指个体对自己控制环境能力的知觉。内控型的人认为他们能够控制环境，并能取得事业上的成功；而外控型的人则认为自己的生活被外部因素（如机遇或者其他权威人物）所控制。内控型的人多使用以问题为中心的应付策略，对组织变革表现出更为积极的态度，较少受应激源的不利影响，对组织变革适应较好。

（2）自我效能感。自我效能感是个体对自己是否有能力采取行动以产生某种特定结果的一种信念。自我效能感可以影响个人应付方式。当个体面临新奇、较难预料或解决的情境（如，变换工作或失业等）时，自我效能感是个体可以利用的一种重要心理资源，它可以激励个体采取灵活的行动去适应环境的变化。

（3）自尊。自尊代表了个体在多大程度上认为自己能干、重要、成功和有价值。高自尊与工作满意度、失业后成功地找到工作以及再就业后的成功等有着密切相关，它对雇员有效地适应组织变革有直接影响。

（4）积极情感。积极情感代表着一种内在的人格倾向，通常表现为健康、自信、精力充沛、合群及归属感等特征。一般地，积极情感高的个体能体察并服从于组织的变革，着眼于问题，与同事和其他人建立积极的人际关系，缓解由变革带来的心理压力。

（5）容忍力。由于组织变革中通常都包含着各种不确定性，所以对不确定性的容忍力就成为影响个体适应组织变革的重要人格变量。容忍力不仅与个体是否容纳和接受新事物有关，还与个体是否愿意改变自己对事物的观点相联系。不确定性容忍力较低的个体在面对组织变革时会表现出角色模糊与心理紧张。

（6）风险偏好。组织变革对个体而言意味着风险，因此个体的风险偏好会直接影响其对变革的态度及相关行为。厌恶风险者在变革情境中常常感到不安，将新奇和风险情境视为否定的，试图脱离这种情境，因而不能成功地应付变革。

第二节　组织发展

一、组织发展的概念、目标及模式

（一）组织发展的概念

组织发展（organizational development,简称 OD）是近年来兴起的一个重要的研究领域，是对组织的某些部分或方面进行修正或整顿，或对整个组织进行有计划的、系统的、长远的调整的过程，致力于改进组织和组织中的人。与组织变革不同的是，组织发展是一种改善或改良的策略，是使组织和组织中的人更加完善的方法。组织发展并没有一个统一且公认的定义，但对这一领域的描述却有很多相似的地方，概括起来有以下几点：

（1）组织发展是一种有计划的努力。一项组织发展计划涉及到对组织的系统性诊断、组织改善的战略规划以及实施规划所需要的资源调配方案等。

（2）组织发展涉及整个系统。组织发展事关整个组织的前途，涉及组织的方方面面，诸如文化、奖励制度以及整个管理战略上的改变等。当然也可能会有一些与组织子部门相关的战术改变，但是整个变革系统是完整的，它是一个相对自治的系统，能自主决定自己的计划和未来。

（3）组织发展是自上而下进行的。在组织发展中，系统的高级管理层对整个计划倾注心血，并承担风险和责任。他们可能不直接参与具体的组织活动，但是他们必须具有控制计划实施过程的能力和达成目标的知识，并兑现有关的承诺，而且必须积极支持那些能够用来实现目标的方式方法。

（4）组织发展的目标是增进组织效能和健康。一个健康的组织有着清晰、可接受的组织目标，相对顺畅的沟通渠道，资源能得以充分利用，人岗良好匹配，并拥有适度的凝聚力，能与环境相互适应。

（5）组织发展运用行为科学知识，通过有计划的干预来实现目标。组织发展的基本策略是改良或干预，而非变革或强制。所使用的干预手段都是从行为科学中提炼吸收而来的，如个人动机管理、文化规范、问题解决、目标设置、人际关系、冲突管理等。

（二）组织发展的目标

作为增强组织有效性和改善组织成员工作生活质量的一种系统工具，组织发展出现在 20 世纪 50 年代。第二次世界大战以后，心理学、组织行为学、组织理论、管理学、人力资源管理等行为学科的发展，使有关组织的知识、研究和理论有了长足的进步，催生了组织发展系列问题的研究。组织发展的主要目标在于提高组织效益和组织的应变能力，具体表现在以下几个方面：

（1）建立一个自我更新、自我完善、自我发展的系统，能够根据工作任务不同而运用不同的组织形式。灵活机动是该系统的典型特征。

（2）通过建立内部持续改善机制，使稳定的（基本组织构架）和临时的（使组织大部分工作任务得以完成的项目组、团队等）系统有机配合，实现效率最优化。

（3）相互依赖的部门或个体之间逐渐走向高合作、低竞争的轨道。妨碍组织效率的主要阻力之一就是将精力消耗在无谓的竞争上，部门之间、员工之间为了某种利益明争暗斗，完成任务时却显得力不从心。如果将这种内耗的精力都用到生产和服务上，那么组织的生产效率和服务质量将会有大幅度的提升。

（4）创造条件使冲突暴露出来并予以管理。在一个不健康或者亚健康的组织中，最基本的一个问题是错误地把精力花在应付、回避或掩盖冲突上，而这些冲突在一个复杂的环境中是不可避免的。组织应该逐渐将冲突视为一种不可避免的情况，创造条件使之暴露出来并予以解决。

（三）组织发展的基本模式

（1）扩大经营规模，提升经营层次。企业组织通过扩大经营规模，采用跨区域乃至国际化经营的方式实施发展。改革开放以来，许多国外的著名企业纷纷在中国设立经营机构和场所，这是扩大经营规模的典型方式。近年来国内的海尔、联想、华为、中兴等优秀企业，也纷纷以跨国经营的方式扩大企业的经营规模，而金融、零售等行业更是在增加经营网点、扩大经营规模方面，提供了无数的现实案例。提升经营层次则主要体现在国内民营企业正在逐步建立起区域或国内总部；许多外资企业进入中国市场后，经营层次逐步由办事处升级为实体经营，进而升级为投资公司。

（2）实行跨行业、多元化经营。虽然对于企业的跨行业、多元化经营，理论界存在着不同的观点和争议，但许多成功的跨国公司几乎都采用或选择了跨行业、多元化经营的策略。多元化是企业寻求新的增长点、降低经营风险的一种有效方式。

（3）实施兼并与收购。近年来，企业组织纷纷采用兼并或收购的方式，实现自身的发

展。企业期望通过并购来扩大经营规模、提高经营效益、争夺市场权力、实行多元化经营等，并期望充分发挥并购后的经营协同效应、财务协同效应、市场份额效应等优势，使并购双方在改善经营、降低成本、提高经济效益、增强技术优势和市场竞争力等方面得到改进。

（4）调整核心业务，拓展新的业务领域。通过调整核心业务，实行经营内涵的重大转型，也是企业组织发展的策略与模式。IBM 是一个很好的例子：作为具有几十年经营历史、以制造计算机硬件及整机为核心业务的电脑业“蓝色巨人”，近年来根据全球计算机行业经营环境的变化及企业利润增长的需要，将其核心业务转到了计算机软件及经营管理咨询服务领域。

二、组织发展过程中的组织公正

公正作为一种道德和价值标准，一直是许多人的社会理想，也是许多学科共同感兴趣的问题。在组织发展过程中，组织规模扩大了，资源更加丰富了，同时资源如何分配的问题也进一步凸显出来。组织中的资源分配，向来是一个很敏感的问题，这种分配是否被组织成员认为是公正合理的，不仅直接关系到他们的满意感、士气与行为表现，还会影响到组织的稳定与秩序及组织发展的目标能否实现。实际上，组织公正不仅是组织发展的重要途径和手段，还是组织发展的重要目标之一。

（一）组织公正概念的界定

对于组织公正的概念内涵，至今没有统一的说法。究其原因，在于公正没有一个绝对的标准，每个人对怎样才算“合情合理、不偏不倚”的看法可能各不相同，公正与否实际上很难统一。因此，目前对于组织公正的界定大多从它的结构维度出发，分三个方面加以分析。

（1）分配公正。分配公正是指对资源配置结果的公正感受。1975 年之前的公正研究主要都集中在这一方面。亚当斯（Adams）认为，公正是人们进行社会比较的结果。人们将自己的结果或收益与自己的投入或贡献（如学历、智慧和经验）的比率与参照对象的这一比率进行比较，若二者比率相等则产生公正感，若比率不相等则会产生不公正感。

（2）程序公正。程序公正更强调分配资源时使用的程序、过程的公正性。研究者发现，当人们得到了不理想的结果时，如果认为过程是公正的，也能接受这个结果。换句话说，假如争执者认为他们能控制做决策的过程（如可以提出自己的证据、有机会表述自己的意见），他们的公正感就会提高。这种现象被称为公正过程效应或发言权效应。

（3）互动公正。互动公正主要关注的是当执行程序时，人际处理方式的重要性。互动公正包括两种：一种是人际公正，反应了在执行程序或决定结果时，权威或上司对待下属是否有礼貌、是否考虑到对方的尊严、是否尊重对方等；另一种是信息公正，主要指是否给当事人传达了应有的信息，即给当事人提供一些解释，如为什么要用某种形式的程序，或为什么要用特定的方式分配结果。

（二）组织发展过程中的组织公正

在组织发展中，要提高员工的工作绩效和工作满意度，促使员工产生和保持对组织的承诺和组织公民行为，吸引并留住企业所需的人员，企业必须在人力资源管理的诸多环节保证组织的公正。这一点在绩效评价和薪酬管理两方面尤其显著。

1．绩效评价中的组织公正

不断改进绩效评价系统是组织发展的重要内容，也是体现组织公正的重要方面。评价结果可为许多其他的人力资源活动如报酬分配（奖金、晋升等）、个人职业发展、培训需求的评估等提供有价值的绩效信息，以增进人力资源决策的有效性和满足员工对于绩效反馈的需求。在绩效评价中，公正主要体现在评价标准的合理性上，即评价标准应该建立在工作分析的基础上，通过对岗位的工作内容、职责、所需的知识的客观度量，确定适当的评价尺度，并严格按照该尺度进行评价。但是，在实际操作中，某些工作很难找到完全客观的评价尺度，主观性不可避免地会掺入进来，而且被评价者情况各异、观点不同，很难做到或根本做不到绝对的公平合理，从而引起部分员工的不公正感。

因比，在评价程序上多做努力便成为弥补评价标准缺陷的一种重要方法。福格（Folger）等人建议采用司法实践中“合法程序”的概念和方法，来实施程序公正。合法程序包括三个步骤：（1）足够的通知；（2）公正聆听；（3）基于证据的判断。

将合法程序运用于绩效评价中，可以采用面谈评价法或称员工谈话制度，以此来代替单纯的测验式或主观性评价。面谈评价法是由员工与其主管就员工的工作绩效进行面谈，做出等级评定，是有关员工工作绩效信息的双向沟通。运用合法程序模型进行绩效评价，结果发现，对于员工来说，即使得到较低的评定等级，也比没有采用合法程序评价的员工更能感受到评价标准的公正、决策信息的准确，对工作和组织的满意度较高，对管理者的评价较高，并更愿意改善自己的绩效和留在组织中；对于管理者来说，他们表示了对评价系统和工作的满意，认为这是解决问题的较好办法，并且报告他们较少因个人观点扭曲评价结果。无疑，评价程序的公开化、透明化是提高员工公正感、增进评价有效性的重要途径。

管理者给予负面的绩效反馈通常会引起员工的不满和拒绝，甚至可能产生工作场所的攻击行为。在管理者不得不给予较低的评价反馈时，人际公正有助于防止这些负面情绪和行为的发生，主要方法有：诚恳地解释为什么给予较低的评定等级，表示诚挚的遗憾，尊重被评定者，避免疏远被评定者。

2．薪酬管理中的组织公正

重新设计和塑造薪酬系统也是组织发展的重要内容，通常需要综合考虑岗位差异、员工个体特征（如资历、各种胜任特征等）和工作绩效等因素的平衡。一些组织采用的点排列法，以及对技术人员实行技术职位序列对应于相应的管理职位序列的报酬标准，都是保持分配公正、防止人员流失的措施。

然而，分配上绝对公正的薪酬体系只能是一个理想化的目标，因而需要采用程序公正和人际公正措施来弥补薪酬体系的不足，保证薪酬管理的有效性。在实践中，以下几个措施是行之有效的：（1）给予员工选择薪酬形式和参与薪酬系统设计的机会；（2）公开组织

的报酬结构和计算方法，使员工了解薪酬决策的程序以判断其合理性；（3）做好有关薪酬结果的沟通；（4）当不得不打破自身的报酬等级而按市场价格招募所需的特别人才时，向其他员工详细地解释原因，使员工从心理上认同这是企业发展的需要。

当前在很多企业流行的保密工资制，在很大程度上是与程序公正和人际公正原则相左的。问题在于，这种工资制度形式本身就向员工传递了不信任的信息，而且容易引起员工的怀疑，产生各种各样的推测或联想。在美国和中国，都发生过实行保密工资制企业的报酬资料被窃取公开，从而在组织内引起轩然大波的事例。一旦发生这样的情况，无论企业的薪酬体系本身是否公正，都会极大地降低员工的满意度和组织士气。因此，如果出于某种原因一定要实行保密工资制的话，保密的也只能是每个人收入的具体金额，而工资的构成要素和计算方法不仅应该是公开透明的，还应该与员工进行充分的沟通，给予相应的解释。这昭示我们：组织发展任重而道远，功夫在平时。程序公正与人际公正是组织发展过程中应加以关注的重要问题。

本章小结：

1．随着知识经济时代的到来，企业所面临的各种环境条件发生了深刻的变化，组织变革成为必然的、唯一的选择。具体而言，生产技术的发展、信息技术的进步、顾客需求的变化、价值创造的新趋势构成了组织变革的外部动因。

2．按照勒温的力场分析模型，组织变革就是组织由目前状态向预期状态的转变过程。在这个过程中，促使组织变革的驱动力与抵制组织变革的抵抗力在不同的水平上展开斗争，取得平衡，逐渐向预期状态过渡。

3．解冻、变革和重新冻结是组织变革三个重要环节。解冻就是引入变革动因，打破现有的组织平衡状态。在这个过程中，组织一方面要设法增强变革的驱动力，另一方面要设法降低变革的抵抗力。变革即按计划实施和推行变革措施，过渡到预期状态。它可能是渐进的、微妙的过程，也可能是突进的、戏剧性的过程。重新冻结即巩固变革成果，防止回到以前的角色和行为模式。

4．制定战略规划、确定适当的变革代理人和推广变革成果是实施和推行组织变革的三大重要举措。

5．组织变革的结果通常体现在组织结构的变化和员工个人的变化上。近年来，通过组织变革出现了柔性动态型组织、结构异化型组织、虚拟网络型组织等新型的组织结构，每种组织结构又各自包含了不同的组织形式。员工个人的控制源、自我效能感、自尊、积极情感、对模糊的容忍力、风险偏好等特征影响其能否很好地适应组织变革带来的变化。

6．组织发展是对整个组织进行有计划的、系统的、长远的完善和调整的过程，是一种改善或改良的策略，是使组织和组织中的人更加完善的方法。其目标在于建立自我更新、自我完善、自我发展的系统，建立内部持续改善机制，使相互依赖的部门或个体之间逐渐走向高合作、低竞争的轨道，使冲突暴露出来并予以管理。

7．组织发展的基本模式有：扩大经营规模，提升经营层次；实行跨行业、多元化经营；

实施兼并与收购；调整核心业务，拓展新的业务领域。

8．组织公正不仅是组织发展的重要途径和手段，还是组织发展的重要目标之一。它包括分配公正、程序公正、互动公正三个不同的维度。绩效评价和薪酬管理是组织发展中实现组织公正的两个重要方面。程序公正和人际公正可弥补分配公正的缺陷或不足。

本章思考题：

1．结合我国企业的实际，谈谈组织变革的必要性。
2．试述勒温的力场分析模型，举例分析该模型对于说明组织变革过程的贴切性。
3．举例分别说明在解冻、变革和重新冻结环节上应注意的问题。
4．试从组织结构角度分析组织变革的结果。
5．个人怎样才能更好地适应组织变革带来的变化？
6．简析组织发展的概念和目标。
7．如何在组织发展中实现组织公正？

参考文献

俞文钊．管理心理学．上海：东方出版中心，2002

朱永新．管理心理学．北京：高等教育出版社，2002

薛振田．管理心理学原理与应用．青岛：中国海洋大学出版社，2005

陈力华，邱羚．组织行为学．北京：清华大学出版社，2005

赵应文，邵继红等．组织行为学．武汉：武汉理工大学出版社，2005

陈向明．质的研究方法与社会科学研究．北京：教育科学出版社，2000

刘永芳．归因理论与人力资源管理．上海：上海教育出版社，2007

周斌．论行为科学、组织行为学与管理心理学的区别与联系．四川教育学院学报 2005，18（5）：25～26

张向葵．管理心理学．长春：东北师范大学出版社，2003

李磊，马华维．管理心理学．天津：南开大学出版社，2006

凌文辁，郑小明，张治灿等．组织心理学的新进展．应用心理学，1997，3（1）：34～36

时勘，卢嘉．管理心理学的现状与发展趋势．应用心理学，2001，7（2）：43～45

刘霞，潘晓良．世纪之交的中国管理心理学:问题与危机．管理科学，1998，21（4）：25～27

吴岩．教育管理学基础．北京：清华大学出版社，2005

JF 利奥塔．后现代状态．上海：三联书店，1997

李铮．后现代思潮与心理学．安徽师范大学学报（人文社会科学版），1999，24（1）：56～58

弗克马，伯顿斯编．走向现代主义．王明等译．北京：北京大学出版社，1992

彼得•德鲁克．后资本主义社会．上海：上海译文出版社，1998

罗珉．西方后现代管理思潮的评论．财经科学，2002，27（3）：21～22

郑航生．现代西方哲学主要流派．北京：中国人民大学出版社，1998

安德列耶娃．西方现代社会心理学．北京：人民教育出版社，1987

高觉敷．西方近代心理学史．北京：人民教育出版社，1982

赫尔雷格尔，斯洛克姆，伍德曼．组织行为学．第 9 版．俞文钊，丁彪等译．上海：华东师范大学出版社，2001

罗宾斯．组织行为学．第 7 版．孙健敏，李原等译．北京：中国人民大学出版社，1997

斯蒂芬•P. 罗宾斯．管理学．黄卫伟等译．北京：中国人民大学出版社，1997

全国 13 所高等院校编写组．社会心理学．第 3 版．天津：南开大学出版社，2003

张克昕．现代管理心理学．理论与应用．北京：航空工业出版社，1998

彭瑞祥．中国劳动心理学三十年．心理学报，1980，3（1）：54～58

徐联仓，王重鸣．管理心理学研究．中国心理科学（工作心理学分卷），1997

赵莉如．中国现代心理学的起源和发展．心理学动态，1992年

尚宇红．博弈论在中国的传播及目前存在的问题．生产力研究，2006，8（7）：16～18

何伟．博弈论与现代企业管理．重庆工商学院学报，2003，19（5）：45～46

陈翠美．管理激励、约束机制设计中博弈论的运用．工商管理，2005，15（6）：32～34

蔡宏标、王陈．从两个博弈论的例子浅谈理性问题．经济师．2006，21（9）：18～20

黄希庭．人格心理学．南京：江苏教育出版社，2002

弗洛伊德．精神分析引论．高觉敷译．北京：商务印书馆，1998

张剑，郭德俊．企业员工工作动机的结构研究．应用心理学，2003，9（1）：3～8

张炼．企事业人员工作动机研究．重庆师范大学学报（自然科学版），2004，21（1）：62～64

德・鲁森斯．组织行为学．王垒等译．北京：人民邮电出版社，2003

Duane，PSchultz & Sydney Ellen Schultz．工业与组织心理学——心理学与现代社会的工作．时勘等译．北京：中国轻工业出版社，2004

刘永芳．归因理论及其应用．济南：山东人民出版社，1998

张爱卿．动机论：迈向二十一世纪的动机心理学研究．武汉：华中师范大学出版社，1999

高觉敷．西方心理学的新发展．北京：人民教育出版社，1987

苏东水．管理心理学．第3版．上海：复旦大学出版社，1998

夏国新，张培德．新编实用管理心理学（修订本）．北京：中央民族大学出版社，1998

林建煌．管理学．上海：复旦大学出版社，2003

徐长江，时勘．对组织公民行为的争议与思考．管理评论，2004，16（3）：45～51

张小林，戚振江．组织公民行为理论及其应用研究．心理学动态，2001，9（4）：352～360

赵曙明．中国企业人力资源管理战略研究．南京：南京大学出版社，2003

斯科特・普劳斯．决策与判断．施俊琦，王星译．北京：人民邮电出版社，2004

斯蒂芬・P. 罗宾斯．组织行为学．第10版．孙健敏，李原译．北京：中国人民大学出版社，2005

R・A. 巴伦，D・伯恩．社会心理学．黄敏儿，王飞雪等译．上海：华东师范大学出版社，2004

Joseph,EChampoux．组织行为学：基本原则．第2版．宋巍巍，张微译．北京：清华大学出版社，2004

肖余春．组织行为学．北京：中国发展出版社，2006

TaylorSE,Peplau,LA & SearsDO．社会心理学．第10版．谢晓非等译．北京：北京大学出版社，2004

MeyersD．社会心理学．第8版．侯玉波，乐国安，张智勇等译．北京：人民邮电出版社，2006

ColvinG．别去指责沃尔玛．未尧译．财富（中文版），2006（3）

卡尔·弗莱保罗．知识管理．徐国强译．北京：华夏出版社，2004

彼得·圣吉．第五项修炼——学习型组织的艺术与实务．郭进龙译．上海：上海三联书店，1998

彼得·圣吉．变革之舞——学习型组织持续发展面临的挑战．王秋海等译．台湾：东方出版社，2001

张洪兰，张晓蓉．现代组织学．上海：复旦大学出版社，1997

赵士英．显性知识与隐性知识的辩证关系．自然辩证法研究，2001，10：20～23

陈国，马萌．组织学习的过程模型研究．管理科学学报，2000，3（3）：15～23

罗彪等．组织学习与理论实施模型．研究与发展管理，2003，15（4）：7～13

赵风中．组织学习障碍探析，科学管理研究．2006，24（1）：84～87

倪文斌等．组织遗忘模式及管理策略研究．哈尔滨工业大学学报，2006，8（3）：73～77

理查德·格里格，菲利普·津巴多．心理学与生活．王垒、王甦等译．北京：人民邮电出版社，2003

巴里·施瓦茨．无从选择——为何多即是少．凌伟文译．北京：中国商务出版社，2005

HempelJ．蓝色巨人掀起头脑风暴．张富贵译．商业周刊（中文版），2006，9：79

王金加，雷刚．二十一世纪的最佳员工福利——员工心理健康管理．当代经理人，2006，9：154～155

何云．情感性劳动的三种类型及其影响．学术研究，2003，3：50～52

潘习龙，何云．情感性劳动中的角色误区．商业研究，2004，288（4）：168～169

文书生．西方情绪劳动研究综述．外国经济与管理，2004，26（4）：13～17

卞冉，龙立荣．工作倦怠的理论研究及其进展．中国临床心理学杂志，2003，11（4）：317～320

徐长江，时勘．工作倦怠：一个不断扩展的研究领域．心理科学进展，2003，11（6）：680～685

李靖．枯竭研究进展．北京大学学报（自然科学版），2003，39（2）：286～290

刘帮成，张海燕．情绪劳动管理：组织与人事管理中的新课题．中国人力资源开发，2003，3：95～98

杰拉尔德·格林伯格，罗伯特·A 巴伦．组织行为学．范庭卫等译．南京：江苏教育出版社，2005

Paul RTimm & Brent DPeterson．人的行为与组织管理．钟谷兰译．北京：中国轻工业出版社，2004

商磊．组织与人格——现代管理中的心理因素．北京：中国政法大学出版社，2006

波特·马金，凯瑞·库伯等．组织和心理契约．王新超译．北京：北京大学出版社，2000

赵辉．企业员工工作价值观的研究．中国优秀博硕士学位论文全文数据库，2005

黄文艳，马剑虹．组织中的心理契约．人类工效学，2006，1：34～36

理查德·哈格斯等．领导学．北京：清华大学出版社，2006

程正方．现代管理心理学．北京：北京师范大学出版社，2004
詹姆斯·库泽斯等．领导力．北京：北京电子工业出版社，2004
加里·尤克尔．组织领导学．北京：中国人民大学出版社，2004
米歇儿·海克曼．领导学（沟通的视角）．上海：上海人民出版社，2004
理查德·L 达夫特．领导学原理与实践．北京：机械工业出版社，2005
冯明．对工作情景中人的胜任特征的研究．外国经济与管理，2001，23（8）：25
时勘，王继承，李超平．企业高层管理者胜任特征模型评价的研究．心理学报，2002，34（3）：306～311
仲理峰，时勘．家族企业高层管理者胜任特征模型．心理学报，2004，36（1）：110～115
姚翔，王垒，陈建红．项目管理者胜任力模型．心理科学，2004，27（6）：1497～1499
梁建春，时勘．组织的核心胜任特征理论及其人力资源管理．重庆大学学报，2005，11（4）：127～129
俞文钊．现代领导心理学．上海：上海教育出版社，2004
于海波，方俐洛，凌文铨．全球性领导者的素质与培训．心理科学进展，2003，11（4）：446～451
彭逼眉．胜任特征模型及其在人才选拔中的应用．武汉大学硕士学位论文，2004
时勘，王继承，李超平．企业高层管理者胜任特征模型评价的研究．心理学报，2002，34（3）：306～311
陆昌勤，方俐洛，凌文铨．360 度反馈及其在人力资源管理中的效用．中国管理科学，2001，9（3）：74～80
陈民科，王重鸣．评价中心的开发程序与构思效度．人类工效学，2002，8（2）：27～34
殷雷．关于评价中心若干问题的探讨．心理科学，2006，29（4）：1007～1009
朱祖祥．工业心理学．杭州：浙江教育出版社，2001
北京新华信商业风险管理有限责任公司译校．谈判与冲突化解．《哈佛商业评论》精粹译丛．北京：中国人民大学出版社，2004
JohnRSchermerhorn,JamesGHunt,& RichardNOsborn．组织行为学．第 8 版．北京：清华大学出版社，2005
王黎，张建新．从人格特质角度看管理人员应付组织变革．心理学动态，2000，8（4）：57～62
刘帆．现代企业组织变革新视野．武汉市经济管理干部学院学报，2005，19（3）：8～12
米旭明，黄黎明．企业组织变革影响因素研究．当代经济管理，2005，27（1）：43～105
李晔，龙立荣，刘亚．组织公正感研究进展．心理科学进展，2003，11（1）：78～84
李绪红．组织公正：从理论到应用．经济理论与经济管理，2002，7：57～61
Alice Lam.*Tacit Knowledge,Organizational Learning and Socieal Institutions:An Integrated*

Framework.Organization Studies,2003，23（3）：487～513.

ArgyrisC & SchonDA.*Organizational Learning:Theory,Method and Practice*.Reading MA: Addison～Wesley,1996，2

BlackJS,GregersenHB.*High impact training: forging leaders for the global frontier*.Human Resource Management,2000，39（summer/fall）：173～184

BongiornoL.*The duller the better:For 1994's annual reports,modesty is a virtue*.Business Week,1995：44

BreenB.*What's Your Intuition*.Fast Company,2000：290～300

DaltonM.*Are competency models a waste*. Training and Development,1997，51：46～49

DemosT.*The World's Most Admired Companies*.Fortune,2006，67～68

DodgsonM.*Organizational learning:a review of some literatures.* Organization Studies,1993，13（2）：25～34

Donna Marie De Caro1is.*Competencies and Imitabi1ity in the Pharmaceutica1 Industry: An Analysis of Their Relationship with Firm Performance*.Journa1 of Manage～ment,2003，29（1）：27～50

DuBoisDD. *Competency modeling*.In：DGLangdon,KSWhiteside& MMMcKenna. *Intervention resource guide:50 performance improvement tools* .San Francisco:Jossey～Bass/Pfeiffer.1999：106～111

Edmondson & Moingeon.*Organizational learning and competitive advantage*.New York: Mc Graw～Hill，1997

GatewoodRD,GowanMA & LautenschlagerGJ.*Corporate image,recruitment image, and initial job choice decisions*. Academy of Management Journal,1993，36：414～427

GreenbergJ & JeraldRA.*Organizational Behavior*. 6th Edition. New Jersey: Prentice～Hall 1996

HouseRJ & SinghJV.*Organizational Behavior*. Annual Review Psychology，1995

JohannessenJA.*Aspects of Innovation Theory Based on Knowledge Management.* International Journal of Information Management,1999，19(2)：121～139

JohnsGA.*Multi-level Theory of Self-Serving Behavior in and by Organizations*.Research in Organizational Behavior,1991，21：1～38

KatzellR & AAustinJT.*From then to now:the development of industrial-organizational psychology in the United States*.Journal of Applied Psychology,1992，28：87～90

KhatriN & NgHA.*The Role of Intuition in Strategic Decision Making*.Human Relations,2000，53：57～86

Lyne～StevensonRM.*Expectancy theory and predicting future employment status in the young unemployed*.Journal of Occupational and Organizational Psychology,1999，72：101～106

McClellandDC.*Identifying competencies with Behavioral Event Interviews* Psychological Science, 1998，9：331～339

McClellandDC.*Testing for competence rather than for intelligence.*American Psychologist, 1973，28，1～14

MirabileRJ.*Everything you wanted to know about competency modeling.*Training and Development Journal,1997,41：73～77

NelsonDL & Quick J C.*Organizational Behavior:Foundations,Realities & Challenges.*3rd Edition.Ohio:South-Western College Publishing，1999

SanchezR,HeeneA ＆ ThomasH.*Towards the theory and practice of competence-based competition.* In:Sanchez R, Heene A, Thomas H.*Dynamics of competence-based competition:theory and practice in the new strategic management.*London:E1sevier,1996：1～35

Schein EH.*Organizational Psychology*,Englewood Cliffs：PrenticeHall，1980

SpencerJrLM,SpencerSM.*Competence at work:Models for superior performance.*New York:John Wiley & Sons，1993

SchmittNChan，D.*Personal selection:A theoretical approach.*A Sage Publication Series, 1998：45～46

Stevens CK & KristofAL.*Making the right impression:A field study of applicant impression management during job interviews.*Journal of Applied Psychology,1995，80：587～606

Weick KE & QuinnRE.*Organizational change and development.* Annual Review Psychology，1999

Williams RS.*Performance management.*London: International Thomson Business Press，1998

互联网+教材

21世纪经济管理精品教材

金融学系列

Behavioral Finance

杨晓兰 张雪芳 等 编著



清华大学出版社

行为金融学

杨晓兰　张雪芳　等　编著

清华大学出版社
北　京

内 容 简 介

行为金融学是一门金融学、心理学等学科交叉的新兴学科，旨在突破传统金融理论对理性人的假设，为真实金融市场提供更为有力的解释。本书系统介绍了行为金融学的基本理论，讨论了行为金融学的主要研究方法，并在部分章节提供了实验指南。本书引入基于中国金融市场真实情况的实证研究和案例分析，以及一些基于中国被试的行为实验结果，加深读者对行为金融理论的理解。此外，本书介绍了行为金融学的前沿研究领域，包括大数据挖掘与行为金融、互联网金融中的行为金融以及神经金融学。本书适合作为高年级本科生和研究生的教材。

图书在版编目（CIP）数据

行为金融学 / 杨晓兰等编著．—北京：清华大学出版社，2019
（21 世纪经济管理精品教材．金融学系列）
ISBN 978-7-302-52712-1

Ⅰ．①行…　Ⅱ．①杨…　Ⅲ．①金融行为－高等学校－教材　Ⅳ．①F830.2

中国版本图书馆 CIP 数据核字（2019）第 063174 号

责任编辑： 左玉冰
封面设计： 李召霞
责任校对： 宋玉莲
责任印制： 刘海龙

出版发行： 清华大学出版社
http://www.tup.com.cn，http://www.wqbook.com
地　　址： 北京清华大学学研大厦 A 座　　**邮　　编：** 100084
社 总 机： 010-62770175　　**邮　　购：** 010-62786544
投稿与读者服务： 010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈： 010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者： 三河市吉祥印务有限公司
经　　销： 全国新华书店
开　　本： 185mm×260mm　　**印　　张：** 17.25　　**字　　数：** 395 千字
版　　次： 2019 年 7 月第 1 版　　**印　　次：** 2019 年 7 月第 1 次印刷
定　　价： 45.00 元

产品编号：063256-01

本教材入选浙江大学经济学院本科生系列优秀教材编著与出版计划

前　言

行为金融学是一门金融学、心理学等学科交叉的新兴学科，旨在突破传统金融理论对理性人的假设，为真实金融市场提供更为有力的解释。从 2012 年起，我和张雪芳老师在浙江大学经济学院本科生教学中开设了《行为金融学》这一课程。在授课过程中，我们选取了海内外一些经典的行为金融教材，从中汲取了丰富的养料。随着教学素材的逐步积累，我们也萌发了编写一本教材的想法，目的在于进一步梳理行为金融的理论脉络和实践应用，为教学过程提供更为丰富的案例、数据和实验指南。

相较于已有教材，本教材在阐述理论的过程中尽可能突出以下几个方面的特点：

第一，突出中国市场背景，围绕行为金融理论，介绍了很多中国市场的情况和案例，其中包括近年来本团队利用中国市场数据、中国实验被试在行为金融领域取得的研究结果，有助于学生通过中国情景、中国元素来加深对行为金融理论的理解。

第二，讲述前沿方向，随着现代技术的发展，行为金融学的研究方法得以更新，理论的应用边界也得以扩大，本教材在各个章节中都介绍了国内外相关研究的最新进展，并在第四部分专门介绍行为金融前沿进展，包括大数据挖掘与行为金融、互联网金融中的行为金融以及神经金融学。

第三，介绍实验指南，我们在《行为金融学》课程实践中将实验方法引入课堂，引导学生参与实验，以激发其探究性学习的热情，取得了良好的教学效果，本教材中介绍了一些相关实验的设计指南，供师生参考。

本教材是集体工作的成果，除了我和张雪芳老师之外，参与本书编写的还有高媚、陈紫晴、林依洋、楼晓霞、石琼旖、童昌希、王伟超、伍韵（按姓氏拼音排列），最后由我统稿。教材的编写过程持续多年得以完成，感谢各位编著者的付出。

本教材在写作过程中，参考了国内外已有教材和相关文献，在此向作者表示衷心感谢，特别是中南大学饶育蕾教授、南开大学贺京同教授、中国人民大学周业安教授，他们主编的行为金融、行为经济学相关教材，一直是我们的重要参考资料。同时也感谢浙江大学经济学院对本教材出版的资助，感谢多年来参加我们课程学习的同学们。最后感谢清华大学出版社及左玉冰编辑高效而仔细的工作。

疏漏之处，敬请读者指正。

杨晓兰

2019 年 1 月

目 录

第二部分　宏观行为金融学

第三部分　行为金融学应用

第1章 导　论

经济学是社会科学的王冠，而金融学无疑是这顶王冠上灼灼发光的宝石。从字面上理解，金融即资金融通；从学科的定义来看，《新帕尔格雷夫金融大词典》对金融学的定义是："金融以其不同的中心点和方法论成为经济学的一个分支，其中心点是资本市场的营运、资本资产的供给和定价，其方法论是使用相近的替代物给金融契约和工具定价。"由此可见，金融学的核心问题是金融资产的定价，而这个价格则影响市场资源的配置。在一个有效的金融体系中，各种金融资产均得以合理定价，市场资源均得以充分、有效地配置。

金融市场如何进行资源配置？举一个科技行业的例子。近年来，随着互联网技术的不断发展和网络用户数量的猛烈上涨，全球科技公司的股价持续攀升，美国五大科技巨头FAAMG(Facebook，Amazon，Apple，Microsoft，Google)在2017年的总市值已经突破3万亿美元。FAAMG的市值超过了2016年世界GDP(国内生产总值)排名第五的英国。如果这种高定价是有效而合理的，越来越多的资金就会被吸引到与此相关的产业，产品创新、产业升级将进一步得到推动，经济体系的产业结构也得到优化。也就是说，通过金融资产的价格信号功能，更多的资金将配置到效率最高的产业。

金融市场优化资源配置的功能有一个假设前提，那就是投资人是理性的，能够对公司股票作出合理的定价。然而，在金融市场的发展历史上，定价偏差屡见不鲜。同样以科技股为例，在2000年网络经济的浪潮中，众多网络公司尚不具备盈利能力和盈利前景，但其股价却被狂热的投资者不断推高，最终导致股价泡沫的破裂，造成3万亿美元账面财富的蒸发、500多家企业破产，对实体经济的发展产生了负面影响。图1.1显示了追踪美国S&P500科技类股票(Technology Select Sector Index)ETF走势，在2000年4月达到高点60，直到2017年10月才重回高点。据《财经》杂志的报道，2000年全球互联网用户不到2亿，而2017年全球智能手机用户在30亿以上，科技公司的基本面发生了根本性的变化。

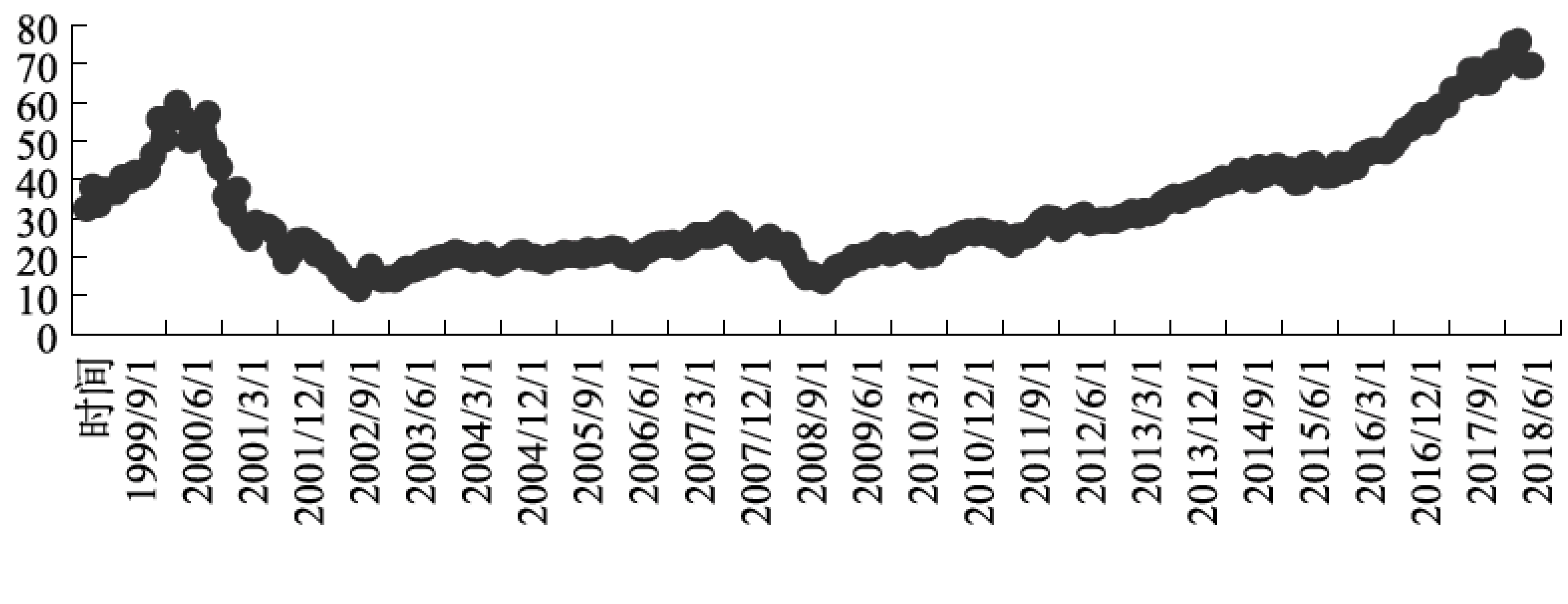


图 1.1　S&P500 科技类股票 ETF 价格走势

长久以来,主流金融学理论以理性人为假设前提,认为投资者能够利用市场信息对股票进行合理估值;即使有部分非理性个体存在决策偏差,市场中聪明的投资者也可以通过套利活动来消除非理性因素的影响,使市场价格保持在完全理性的水平。这就是现代金融理论的基石、有效市场理论的核心观点——股票价格已经包含所有信息,市场的定价是有效的、合理的。例如,假设有投资者错误高估了股票价格,理性的投资者能够立刻根据市场信息发现这一偏差,他们可以通过卖空使股票价格下跌,回到合理的水平。芝加哥大学教授罗伯特·席勒(Robert Shiller)在一篇论文中估算了美国股票市场历史的理性最优价格,发现长期以来股票价格都偏离了理性价格水平,如图 1.2 所示。席勒从心理学的角度来解释这种偏差,他用"动物精神"描述投资者反复无常的态度变化,与理性人假设存在本质的差别。席勒的研究结果在主流金融学界掀起了轩然大波。引入心理学方法的"行为金融学"和坚持理性人假设的"主流金融学"开始了持续数十年的大辩论。

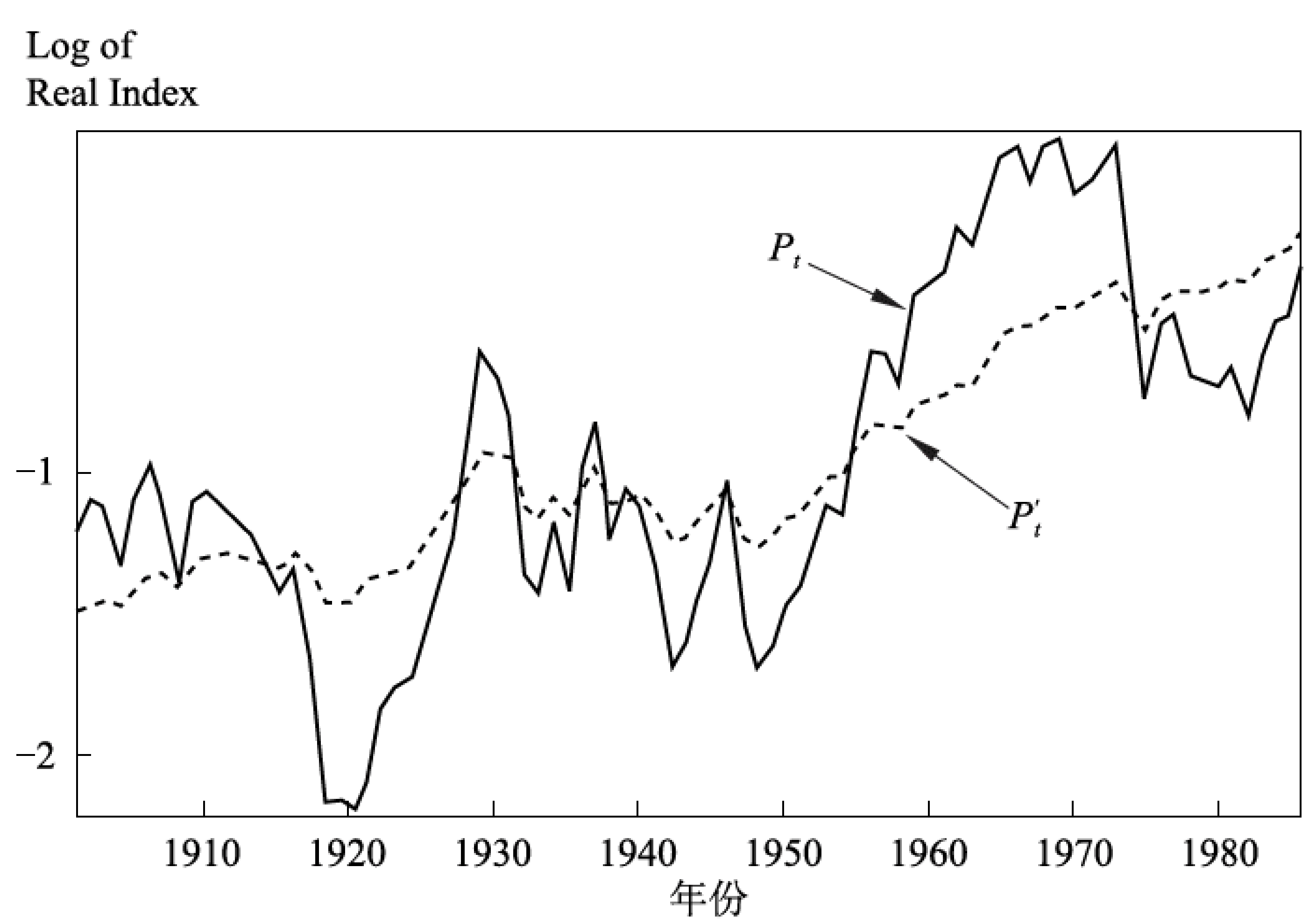


图 1.2　美国股票真实价格 P_t 和理性最优价格 P'_t 的对比

资料来源:Campbell J Y,Shiller R J. *Stock prices,earnings and expected dividends*[J]. *Journal of finance*,1988,43(3):661-676.

随着时间的推移,越来越多的证据揭示了个体的非理性偏差和市场的非有效。例如,同一家公司如果拆分成两家完全相同的公司在不同市场上市,股票价格会存在差异,并且

在长期内也无法消除，荷兰皇家和英国壳牌公司正是这样的例子。再如，封闭式基金通常以每股1元进行发行，发行价格等于基金的净值。然而在上市之后，基金价格却长期低于资产的净值，这被称为“封闭式基金折价之谜”。这些“异象”对有效市场假说提出了挑战，是有效市场理论所无法解释的。

在我国股票市场，投资者的非理性行为也比比皆是。如表1.1所示，看到以下三组股票的名称，会不会认为每组中的两个股票有一定的联系呢？我国A股市场中有很多这样名称类似、实际上却没有任何联系的股票，但投资者往往简单地根据名称就误以为它们之间存在联系。李广子等(2011)的研究发现，如果两只股票名字类似，它们的股价会存在一定的联动性。例如，当人们发现“金马股份”上涨时，会关注到“金马集团”，金马集团的股票也随之上涨，虽然两家公司没有任何关系。另一个例子是我国股民的“数字崇拜”。赵静梅、吴风云(2009)专门研究了我国股票市场中代码尾号为8的股票，发现投资者会追捧这些有发财号码的股票，以8为尾号的股票在上市首日和上市后12个月内的市盈率都远远高于其他尾号组合的股票。

表1.1 名字类似的股票

组别	股票名称	地区	行业
1	金马股份	安徽	制造业
	金马集团	广东	信息技术
2	安泰集团	山西	炼焦业
	安泰科技	北京	制造业
3	北方股份	内蒙古	制造业
	北方国际	广东	建筑业

普遍存在的各种市场“异象”不断地激发人们从多个角度进行新的思索，并对有效市场假说的理论缺陷寻找改进的方法。将心理学与传统金融学相融合的行为金融学由此得到了蓬勃发展。到2017年，已经有卡尼曼(Danial Kahneman)、席勒、塞勒(Richard Thaler)等多位行为金融学家问鼎诺贝尔经济学奖，行为科学在金融研究领域的应用已经得到了广泛的认可，展现出强大的发展前景。本章将介绍行为金融学的发展历史、研究方法和主题。

专栏1.1 行为经济学大师——理查德·塞勒

2017年的诺贝尔经济学奖授予了芝加哥大学的行为经济学大师——理查德·塞勒。塞勒1947年出生于美国新泽西，父亲是一名保险精算师。1974年，他获得罗切斯特大学博士学位，先后执教于罗切斯特大学、康奈尔大学，1995年至今任芝加哥大学商业研究生院行为科学与经济学教授。

和一般的经济学教授喜欢用艰涩的语言论证高深的理论不同，塞勒致力于研究人们在生活中、经济决策中常犯的“错误”。他擅长用诙谐幽默的方式将这些“错误”娓娓道来，引人入胜，发人深思。这里的“错误”是相对传统经济学而言，传统经济学假设个体有能力作出完全理性的决策，也就是“最优的、正确的”决策，但事实并非如此，人们常常犯错。例

如，我们计划控制体重，却禁不住美食的诱惑；我们一次次在股票投资中亏损，却深信自己投资能力高于一般人，亏损只是运气不好。

“心理账户”是塞勒从 1979 年至 1985 年持续研究的主题，是指人们会根据资金来源和用途等因素对资金进行心理上的归类，也就是说“钱和钱不一样”。例如，从股票投资中挣到的收益和固定的工资收入，虽然都是钱，但人们在心里会把这两类钱放到不同的账户里，对于在股票账户里的钱，人们更倾向于将其用于风险更大的投资活动。

1.1 行为金融学的定义和发展历史

1. 行为金融学的定义

作为一门新兴的学科，行为金融学并没有形成一个统一定义。Statman(1987)将行为金融学定义为一种有关在不确定性下如何进行选择的描述性理论。Olsen(1998)认为行为金融学是科学的一部分，它是从基础的原理出发，探索建立在此原理上的能解释金融市场行为的理论。Thaler(1993)则给出了更为宽泛的定义，称行为金融学为“简单地开放思维的金融学(simply open-minded finance)”。尽管对行为金融学的定义尚未达成一致的意见，但是人们普遍认同行为金融学是金融学和心理学的交叉学科，行为金融学结合心理学有关个体认知决策的研究成果，放宽了理性假设的约束条件，从更加接近真实个体的角度来分析现实生活中的投资者行为。

2. 行为金融学的发展历史

从 1571 年伦敦皇家交易所成立以来，金融市场中参与者的心理行为对市场运行的影响一直是投资者特别是投机者关注的重要因素。1852 年，Mackay 在《非同寻常的大众幻想与群众性癫狂》一书中详细记载了密西西比计划、郁金香投机潮及南海风波等金融危机的历史。该书是记录金融市场中由群体行为而导致金融危机的经典作品。此后，该类历史记载性的文献不断增多。1978 年，金德尔伯格(Kinderberg)在其首版的《金融危机史》中对历史上由个体行为引发的金融危机的事件进行了系统而深刻的总结研究。在经济学领域，马歇尔、凯恩斯等经济学家均十分重视经济与市场运行中心理和行为因素的作用，凯恩斯曾认为市场价值在某种程度上由群众心理决定。然而，这些作品却仅限于描述性的思考。

将心理学研究与金融学研究进行科学结合的早期工作来自 Burrell，他于 1951 年发表了 *Possibility of an Experimental Approach to Investment Studies*。在该文中 Burrell 提出了用实验方法进行投资研究的可能性。其后，1967 年，Scott 发表了 *Scientific Investment Analysis：Science or Fiction*？Burrell 和 Scott 共同提出开辟新的将定量投资模型与行为研究方法相结合的金融研究领域。此后，心理学家 Slovic 对投资过程的行为方式进行了系统性的科学探索，Slovic(1972)正式发表了个体是如何对风险产生错误感知的研究成果。

较为公认的行为金融学领域的奠基性工作来自两位心理学家特沃斯基(Tversky)和

卡尼曼。特沃斯基和卡尼曼在1974年的研究中提出了个体决策过程中的启发式(heuristic)偏见,在1979年提出了前景理论(prospect theory),并研究了个体决策中的框定(frame)效应。这些研究对行为金融学的发展产生了深远的影响,可以说从此正式确定了行为金融学的成立。

自20世纪80年代,特别是进入90年代后,行为金融学获得了迅猛的发展。金融学家结合心理学对个体行为决策的研究成果在行为金融学领域进行了大量的探索性研究。从而使行为金融学获得了日趋广泛的影响力。

3. 行为金融学的理论体系

在对传统金融理论进行挑战的同时,行为金融学理论以心理学中关于投资者系统性决策错误形成机制为出发点,对金融市场系统性异常现象(anomaly)进行了解释,逐步构建起自己的理论体系。到目前为止,学者普遍认同行为金融学的研究应该包括微观和宏观两大研究主题,如图1.3所示。



图1.3 行为金融学的两大研究主题

第一个主题是微观行为金融学(behavioral finance micro,BFMI),主要是基于认知和决策相关的视角来研究投资者的心理偏差。行为金融学理论强调投资者是受认知能力限制的,而并非是完全理性的。这一领域的研究与心理学的结合最为紧密,集中探讨了个体决策如何偏离了经典金融学的理性人假设。

第二个主题是宏观行为金融学(behavioral finance macro,BFMA),主要利用市场宏观数据,对有限套利进行研究。行为金融学理论认为由于市场套利力量受限,市场并非是完全有效的,也就是所谓的无效市场模型。这一领域的研究主要探讨金融噪声理论以及引入心理和行为因素的资产组合与定价问题。

围绕着微观行为金融学和宏观行为金融学这两个主题,行为金融学研究金融市场中投资者的心理与行为,研究和探讨在引入心理和行为因素后,如何对传统金融学理论和模型修正与改进,以使其更加符合实际和更具有解释力。正如Statman(2014)的观点,行为金融学是用普通的人(normal people)代替标准模型的理性人。行为金融学最大的理论和实践意义在于,它为金融市场的运行和金融主体的活动提供了内在的心理与行为机制解释,也就是说使金融学的理论体系具备了一个合理和完善的微观基础,不仅可以解释和说明现实金融运行中出现的所谓"异常"现象,如封闭式基金折价之谜、红利之谜、期权之谜等,还给投资者提供了一种新的证券投资的分析方法——行为投资分析方法,同时也为证券市场的监管与政策的制定提供了理论依据。

1.2 行为金融学的主要研究方法

作为一门交叉学科,行为金融学的研究范式与标准金融学存在较大的差异。标准金融理论以理性经济人为假设前提,基本研究方法是通过严谨的计算和推导,从最优化决策的角度出发,建立金融资产定价模型。行为金融学则把完全理性的经济人还原成真实的、有决策偏差的个体,从个体认知的角度出发,理解真实个体的思维决策过程及其对金融资产定价的影响。为了描述真实个体的决策行为,行为金融学在发展初期的研究方法主要来自心理学,包括观察方法、实验方法等,如卡尼曼和特沃斯基的前景理论就完全建立在实验基础之上。随着包括行为金融学在内的行为经济学的不断发展,经济学家探索出在经济学领域进行实验研究的方法,系统性创立了有关经济学实验的新范式,并逐步与心理学实验相区别,形成实验经济学这一相对独立的研究方法,进一步推动了行为金融学领域的理论创新。除采用实验方法之外,更多的经济学家用他们所擅长的计量经济学方法对金融市场的数据进行探究,论证市场存在的种种异象,从而对有效市场理论提出挑战。目前,在计算机科学、神经科学的发展背景下,大数据挖掘、神经实验等前沿方法也已经应用到金融市场的行为研究领域。

从行为金融学的两大研究主题来看,计量经济学、计算机大数据挖掘方法主要应用在宏观行为金融学领域,即利用宏观数据发现套利的有限性,为证明市场非有效性提供证据;心理学、实验经济学、神经科学实验等方法主要应用在微观行为金融学领域,旨在描述、解释投资者的非理性行为,如图 1.4 所示。

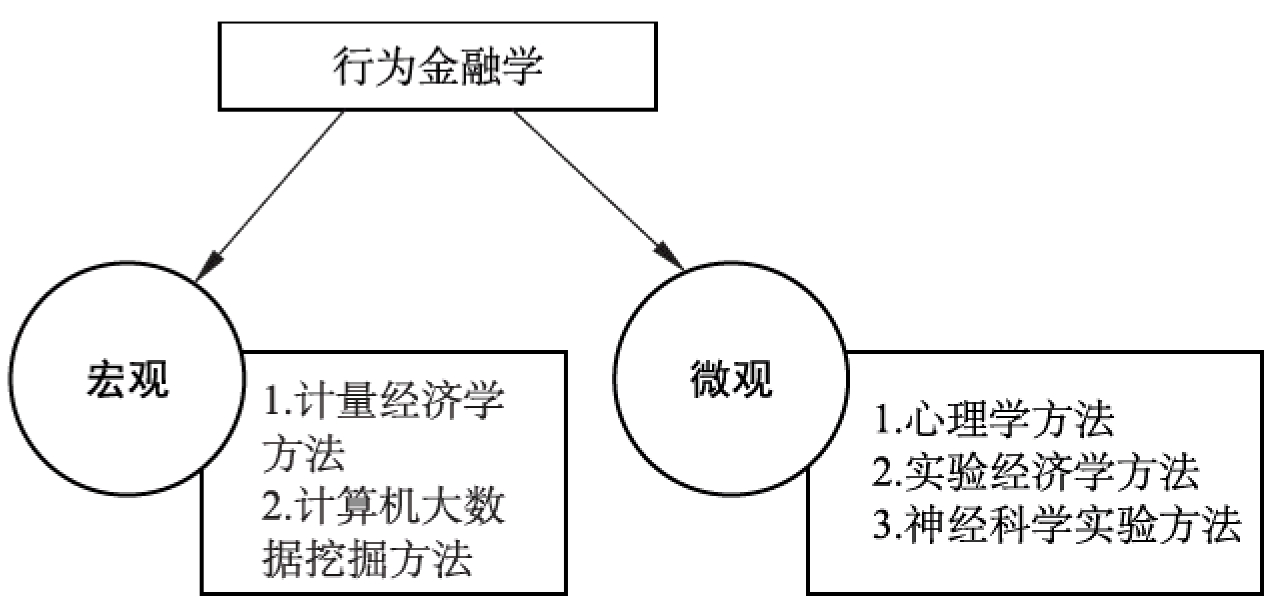


图 1.4　行为金融学主要研究方法

本节将简单介绍行为金融学常用的心理学方法以及实验经济学方法。本书第 13 章和第 15 章将分别介绍计算机大数据挖掘方法和神经科学实验方法。

1. 心理学方法

心理学研究方法可具体分为六大类:观察法、实验法、模型法、测验法、统计法和其他研究方法[①]。与行为金融学理论进展密切相关的三类方法是实验法、测验法和问卷法。

---

① 对心理学方法的系统论述可以参考:王重鸣. 心理学研究方法[M]. 北京:人民教育出版社,2001.

1）实验法

从科学意义上，“实验”是指在确定的条件下，如实验室中，检验某种科学结论或者假设的活动。在科学发展的历史上，实验成为人类认识世界、解释世界的重要工具，它是科学理论的源泉，也是检验科学真理的标准。实验法不仅帮助研究者确认或者推翻一些理论，甚至还可能改变人们对生命、对世界的常识性看法，粉碎存在已久的主观臆断的理论。

然而，在相当多的学科里，实验成为科学研究的重要工具都经历了较长的演进过程。2 000多年前的亚里士多德时代，甚至是物理学都被视为不可实验的科学。400多年前，以伽利略为代表的开创者在物理学中建立了在控制性条件下进行实验的传统。随后，实验方法在化学中的应用也逐步展开。在很长的时间内，生物学被视为是不可实验的，因为人们认为有生命的生物是不可能被作为实验对象的。直至19世纪孟德尔、巴斯德等生物学家才在生物学研究中引入实验技术，而现代生物学基本上是一门实验性的科学了。

心理学，也曾经被认为是无法进行实验的。自冯特创立科学心理学起，现代心理学才跨入实验科学的行列。在近百年的历史进程中，心理实验起初仅用来研究简单的心理反应（如感觉反应、运动反应、选择反应等）和确定各种通道的感受性的阈限。接着，实验被运用于研究记忆过程、知觉过程、注意过程等。目前，心理实验已被运用于心理学的各个分支学科，心理学也体现出明显的实验特征。

在心理学实验中，研究者可以积极干预被试者的活动，创造特定条件使某种心理现象得以产生并重复出现。实验法有两种：实验室实验和自然实验。实验室实验是借助专门的实验设备，在对实验条件严加控制的情况下进行的。运用这种方法有助于发现事件的因果联系，并允许人们对实验的结果进行反复验证。实验室实验的优点是由主试严格控制实验条件，使人们对实验的结果可以进行反复验证。其缺点同样是由主试严格控制实验条件，实验情境往往带有极大的人为性质，往往使实验情境的自然真实性受到损害，从而削弱了研究的外部效度。自然实验也称现场实验，是在人们正常学习和工作的情境中进行的，其结果比较合乎实际。但在自然实验中，自然观察虽然强调真实性，但却以牺牲严密性为代价，且只能反映变量间的相关关系，在对变量的因果关系解释上存在局限性。虽然其外部效度较高，但内部效应却较低。

心理学家费歇尔（Fisher）创造性地把以下三项程序引入心理学实验研究，成为心理学实验研究的重要范式：第一，应用随机化程序选择被试；第二，对被试进行实验控制；第三，运用统计学方法检验研究的结果。

以卡尼曼和特沃斯基的前景理论实验为例，他们随机招募了一些被试，把这些被试随机分成两组，第一组回答问题1，第二组回答问题2。

问题1：请在以下两个选项中作出选择。

A. 100%可以得到100美元。

B. 有50%的机会可以得到200美元，有50%的机会什么也得不到。

问题2：请在以下两个选项中作出选择。

A. 100% 会损失100美元。

B. 有50%的机会损失200美元，有50%的机会什么也不会损失。

卡尼曼和特沃斯基发现在第一个场景下，大部分人会选择 A，而在第二个场景下，大部分人会选择 B。根据这一实验结果，他们得出的结论是人们在赚钱和亏钱两种情况下对待风险的态度存在差异：在赚钱的时候，更倾向于风险厌恶；在亏损的时候，更倾向于风险偏好。本书第 4 章会详细介绍建立在这个实验基础之上的前景理论。

在微观行为金融学领域，有关个体决策偏差的相当多理论都是建立在实验室实验的基础之上的。自然实验虽然具备外部有效性的优势，但在实施上比较困难，在金融领域的应用还比较有限。

2）测验法

测验法是指用一套预先经过标准化的问题（量表）来测量某种心理品质的方法。心理测验（mental test）按内容可分为智力测验、成就测验、态度测验和人格测验；按形式可分为文字测验和非文字测验；按测验规模可分为个别测验和团体测验等。

心理测验要注意两个基本要求：测验的信度和效度。信度是指一个测验的可靠程度。如果一个测验的可靠程度高，那么，同一个人多次接受这个测验时，就应该得到相同或大致相同的成绩。效度是指一个测验有效地测量了所需要的心理品质，它可以通过对行为的预测来表示。

在行为金融学研究中，借鉴心理学量表可以对个体心理偏差，或者某个方面的心理特征进行测度，从而能进一步探索这种偏差、特征对金融决策行为的影响。例如，在行为金融学中一个广受关注的心理特征是“过度自信”，即个体对自己的知识和能力有过高的估计，过度重视自己所拥有的信息而忽略市场上或者他人的信息，或者认为自己能够掌控未来的事件。过度自信会导致股票市场的过度交易、公司决策中的过度投资等非理性行为。个体过度自信的程度可以通过心理学量表进行测量，如采用置信区间问题任务法。一般任务中会设置 10～20 道题目，被试会被要求就一些常识性的问题给出 90％的置信区间。90％置信区间的含义是被试认为正确的值有 90％的可能性落入他所给出的区间，若正确答案包含于该区间内，则认为被试回答正确。

例如，让被试给出“爱因斯坦提出狭义相对论的年份”的区间下界和上界各是多少。在这道题目中，被试并不需要给出爱因斯坦提出狭义相对论的精确年份，只要给出 90％置信水平的下界和上界，如果精确年份落入这个区间，就算回答正确。任何略有常识的人都会知道相对论是 20 世纪最伟大的成就之一，可以给出下界是 1900、上界是 2000 的答案，以使“爱因斯坦提出狭义相对论的年份”落入这个区间。如果被试连相对论出现在 20 世纪这个常识都不具备，甚至可以给出“0～2000”的区间。然而，如果被试对自己的知识有过高的错误估计，就会给出一个更小的区间，如下界 1930 和上界 1950。“0～2000”的答案是正确的，而“1930～1950”这个答案是错误的，因为狭义相对论提出的年份是 1905 年。被试过度自信的程度越高，给出的区间就越小，答案正确的可能性也越小。通过统计被试在 20 道题目中有多少道题给出了正确的区间，就能在一定程度上度量其过度自信的程度。也就是说答对的题越多，过度自信程度越低。

3）问卷法

调查问卷是由研究者提出一些假设的具体问题，被调查人针对这些问题进行回答。

调查问卷也是获得实证数据的一种方法。人们所熟悉的消费者信心指数、经济学家信心指数、景气指数等反映个体预期的指数往往都是通过调查问卷得到的。

由于组织大规模问卷调查的成本较高,研究者也可以从公开的数据库中获取自己需要的变量。与行为金融学相关的数据库主要是微观调查数据库。目前我国的多个机构均开展了大规模的家庭和个人调查,其中涉及家庭支出、金融资产的部分都可以成为探究个体投资者行为的有效素材。例如,李涛、郭杰(2009)利用问卷调查数据研究了投资者的社会互动和风险偏好对股市参与度的影响,他们发现居民的风险态度对其是否投资股票没有显著影响,其原因在于社会互动可以不同程度地降低居民对股票投资风险的主观感知程度,这使得居民的风险态度在解释其是否投资股票时的效果不再显著。胡枫、陈玉宇(2012)利用中国家庭追踪调查(CFPS)数据库研究了社会网络对农户借贷行为的影响。

专栏1.2　常用中国微观调查数据库

(1) 北京大学中国社会科学调查中心:中国家庭追踪调查,主要调查主题是中国居民的经济与非经济福利,包括经济活动、教育成果、家庭关系与家庭动态等。

(2) 西南财经大学中国家庭金融调查与研究中心:中国家庭金融调查(CHFS),主要调查主题是家庭收入储蓄、非金融资产、金融资产、财富、保险等。

(3) 中国人民大学社会学系、香港科技大学社会科学部:中国综合社会调查(CGSS),主要调查主题是社会流动、幸福感、社会信任、教育回报等。

(4) 北京师范大学中国收入分配研究院:中国家庭收入调查(CHIP),主要调查主题是教育回报、收入分配、消费储蓄等。

(5) 浙江大学社会调查中心:中国家庭大数据库(CFD),主要调查主题涉及中国农村家庭比较完整的信息,包括家庭的基本结构、财富、收支、金融行为、人口迁移等。

调查问卷存在的一个局限性是在有些情况下,调查数据难以反映被调查者的真实想法。例如,在一些有关投资者收入与投资行为的调查研究中,被调查者往往不会透露自己的真实收入水平,而且也可能在回答问题的过程中刻意掩饰自己非理性的投资行为。

2. 实验经济学方法

在经济学中应用实验方法最初是从心理学研究中借鉴而来的,行为金融学的奠基人——卡尼曼、特沃斯基本身就是心理学家。然而,科学意义上的实验经济学方法是指由经济学家佛农·史密斯(Vernon Smith)开创的、有规范性准则的用于检验经济学理论的方法。史密斯(1994)将实验经济学定义为:在有显性或者隐含规则的背景下应用实验方法来研究人类相互作用的决策行为。具体而言,实验经济学是在可控制的实验环境下,针对某一经济理论或者经济现象,通过控制某些条件、观察决策者行为和分析实验结果,以检验、比较和完善经济理论并为政策决策提供依据。

与心理学实验相比,实验经济学实验有两个突出的原则:第一是有效激励被试的原则,第二是无偏性的原则。

心理学实验允许给予被试固定报酬,甚至招募获取零报酬的自愿者。实验经济学则强调被试在实验中要获得有效的激励,即需要在实验中使用报酬手段诱发经济主体的特

征，并且应当满足三个条件（Friedman 和 Sunder，1994）：第一，单调性（monotonicity），即实验被试认为报酬量越多越好，报酬不存在饱和状态。第二，凸显性（saliency），被试所得到的报酬必须与他以及其他被试的行动有关，必须由被试理解的制度所决定。第三，占优性（dominance），在实验中实验被试的效用变化来自实验报酬，除此之外的其他原因可以忽略不计。实验经济学家在报酬问题上遵循这一原则，是因为对被试提供报酬能够减少其行为的变异性。基于这样的原因，大部分经济学实验都采用实际的货币报酬支付给被试（也有些实验用咖啡杯、巧克力等商品作为报酬）。货币报酬具有的优势是能够使不同的个体对报酬媒介态度的差异最小化，如果是货币以外的其他商品，不同被试对这些商品的价值评价是非常不一致的，而且货币支付是高度可分的，具有非饱和性的优势。

实验经济学强调无偏性原则。为了达到实验目的，对被试的真实行为进行研究，研究者应该在实验过程中持有这样一种态度，那就是不要使被试产生哪种行为模式是正确的或者是被期望的预期，除非外在的建议也作为一个处理变量。这里的无偏性，不是指研究者不能给出任何明白的行为建议，而是要特别注意避免那些微妙的行为建议。和其他观察实验数据不一样（如化学实验或者物理实验），人作为实验的被试能够按照实验者的要求去做，因此实验者不能通过语言或其他方式暗示什么是好的或者不好的结果。特别是在课堂里进行实验时，有些学生作为实验的被试可能会在实验过程中按照教师所期望的方式去进行选择，希望能给教师留下好的印象，或者得到较高的考试成绩。在这种情况下，作为实验组织者的教师要尽量避免自己的语言对被试的行为产生额外的干扰作用。

实验经济学方法的发展主要体现在三类实验之上。

第一类是市场实验。市场实验思潮以 Chamberlin（1948）为代表。他关心的是产业组织问题，因此他的实验以模拟真实市场交易为目标。Chamberlin 首次提出，实验经济学的目标就是严格剔除无关干扰变量，观察实验被试在特定环境下的真实经济行为。在 Chamberlin 的基础之上，史密斯展开了一系列金融市场实验，探索个体在金融市场的交易行为如何系统性地偏离理性的轨道，如何导致市场价格泡沫膨胀，以及如何设计市场制度以提高市场的有效性。本书第 11 章将介绍这些与金融市场泡沫研究有关的实验。

第二类是个体选择实验。个体选择实验起源于心理学，通常涉及不确定的期望或者是彩票（lottery）问题。彩票的奖金是一个简单的概率分布，如抛硬币，出现正面，则获得 2 美元；出现反面，获得 1 美元。实验被试在两个结果之间进行选择，从而决定他的收益。在早期研究所采用的极其简单的决策环境下，人的行为似乎符合期望效用理论。然而，即使只在决策环境中引进少量的复杂因素，实际行为与期望效用理论的预测之间也立即明显地出现了种种背离。实验结论使期望效用理论面临严峻的挑战，其中比较著名的三类实验包括阿莱（Allais，1953）设计的有关“阿莱悖论”的实验、Lichtenstein 和 Slovic（1971）发展的有关“偏好逆转”的实验以及特沃斯基和卡尼曼（1979）有关“前景理论”的实验，这些实验构成了行为金融学发展的心理学基础。

第三类是博弈论实验。博弈论实验是由心理学家、博弈理论专家和经济学家在 1950—1960 年发展起来的。他们最初的兴趣都来源于著名的“囚徒困境”行为，这个问题是由 Tucker（1950）首次明确提出来的。囚徒困境、最后通牒、公共产品是实验方法在博弈

论领域运用得最多的三个方向。随着博弈理论在经济学研究中的地位日益显著，实验方法的运用也日益受到重视，这是因为实验方法能够用于检验博弈对局人的真实行为，对博弈理论的预测能力和适用范围进行评价和界定。其中与行为金融学关系密切的一类实验是信任博弈或者说投资博弈(trust game/investment game)。这个实验是研究信任、互惠偏好、利他偏好的一种博弈范式，标准的信任博弈范式中，双人分别扮演委托人和代理人角色并拥有初始禀赋 S，委托人可以从初始禀赋 S 中选择投资 $y(0\leqslant y\leqslant S)$ 给代理人，一旦委托人决定投资，总投资额扩大为原来的 N 倍，即代理人自动获得 $N\times y(N>1)$，代理人可以选择返还一个 $x(0\leqslant x\leqslant N\times y)$ 给委托人，此时委托人收益为 $S-y+x$，而代理人收益为 $N\times y-x$。该博弈范式可以用于研究金融市场的借贷行为和投资行为。

专栏 1.3　常用经济学实验软件工具包和平台

(1) z-Tree：z-Tree 程序是由苏黎世大学 Fischbacher 教授开发的实验经济学软件包，在实验经济学教学和研究中广泛应用。实验者需利用该软件包编写程序，通过联网的计算机实现实验过程。

(2) Veconlab：该平台由弗吉尼亚大学 Charles Holt 教授开发，实验者注册后可以直接在线组织实验。可参考 Holt 的书《市场、博弈和策略行为》(上海人民出版社 2011 年版)，该书介绍的相当多实验都可以在 Veconlab 上实现。

1.3　本书的主要结构

为系统性介绍行为金融学的主要内容及其进展，本书分四个部分，共 15 章。第一部分是微观行为金融学，也就是与心理学的结合。第二部分是宏观行为金融学，主要介绍证券市场非有效性的相关理论进展和实证发现。第三部分是行为金融学应用，用过度自信、过度反应、情绪等认知偏差来解释证券市场的定价偏差。第四部分是行为金融学前沿，介绍行为金融学理论在大数据、互联网金融、神经金融等前沿领域的进展。

第 1 章导论。介绍行为金融学的基本概念和方法。

第 2 章行为金融的心理学基础。介绍了判断与决策中的认知过程，以及四种重要的判断启发式及其可能导致的偏差。

第 3 章投资者心理与行为偏差。从人类心理角度，分析投资者在面临不确定性条件下的决策过程中出现的心理和行为偏差，主要包括禀赋效应、短视偏差、证实偏差、后悔厌恶、模糊厌恶、心理账户、时间折扣等偏差。

第 4 章期望效用理论与前景理论。在了解期望效用理论的基础上，主要讨论心理学实验对期望效用理论提出的挑战、前景理论的主要内容、价值函数和决策权重函数的主要特征。

第 5 章并非有效的市场。介绍行为金融学理论对有效市场理论的挑战。在回顾有效市场理论的基本假设和内容之后，主要介绍有效市场理论的局限性，特别是套利的局限性。

第 6 章介绍证券市场的典型异象，主要有股权溢价之谜、封闭式基金折价之谜、日历

效应、规模效应、账面市值比效应、投资者关注度、本地偏好和无信息反应效应等。

第 7 章过度自信，主要介绍个体过度自信的表现形式和测度方法、投资者过度自信对证券市场的影响以及对公司金融决策产生的影响。

第 8 章过度反应，从过度反应的界定、过度反应的心理学解释、过度反应的理论模型、过度反应的实证检验四个方面详细介绍过度反应及其对金融市场的影响。

第 9 章投资者情绪，首先介绍投资者情绪的具体概念、总结投资者情绪度量的方法，其次从理论的角度介绍了投资者情绪影响股票市场的渠道，最后是投资者情绪影响股票市场的实证结果。

第 10 章金融市场羊群行为，介绍羊群行为的心理学原因，以及经济学解释，并讨论如何对羊群行为进行实证和实验检验。

第 11 章金融市场价格泡沫，介绍了金融市场泡沫的定义，重点总结泡沫产生的原因，泡沫测度的方法，讨论了利用实验经济学研究市场泡沫的基本方法。

第 12 章行为公司金融，本章从投资者非理性、管理层非理性两个维度，介绍了非理性行为对公司融资、投资、股利决策、并购决策的影响。

第 13 章大数据挖掘与行为金融，主要介绍大数据在投资者关注、投资者情绪、社会互动、个人信用风险评估等行为金融学研究中的应用。

第 14 章互联网金融中的行为金融，介绍互联网金融的主要模式及发展现状，并根据现有的研究成果，分 P2P 网贷、众筹、亲社会网贷与其他互联网金融模式四块，介绍不同互联网金融模式下的行为金融学现象。

第 15 章决策、不确定性和大脑：神经金融学，介绍神经科学在金融学研究的应用情况，主要是不确定条件下决策行为的神经机制，以及和情绪相关的神经机制研究等方面的进展。

专栏 1.4　行为金融学的代表人物

代表人物	代表性作品/理论
Simon(西蒙)	有限理性
Mackey	《群体》《非同寻常的大众幻想和群体性癫狂》
Keynes(凯恩斯)	“选美竞赛”理论和“空中楼阁”理论
Burell	《一种可用于投资研究的实验方法》
Bauman	《科学投资分析：是科学还是幻想?》
Slovic	《人类判断行为的心理学研究》
Kahneman；Tversky(卡尼曼和特沃斯基)	前景理论
Thaler(塞勒)	心理账户、损失厌恶
Shiller(席勒)	《非理性繁荣》《动物精神》
Shefrin；Statma	资本资产定价理论(BAPM)，行为组合理论(BPT)
Shleifer(史莱佛)	《并非有效的市场》

注：常用代表人物在本书各章中将采用中译名

思考题

1. 行为金融学的研究领域包括哪两个方面？请讨论这两个研究分支的联系和差异。
2. 请简述行为金融学的主要研究方法，比较各种方法的优势和局限性。
3. 请列举金融市场中观察到的非理性现象。

第一部分

微观行为金融学



第 2 章

行为金融的心理学基础

引导案例：市场情绪周期

在西方，人们常说推动股市变动的力量有两个：一个是贪婪，另一个是恐惧。无论贪婪还是恐惧，实际上都是投资者的情绪在影响股市。在股票市场上存在着“情绪周期”(emotional cycle)，《洛杉矶时报》用图 2.1 来演示股票市场的情绪周期和主宰每一时期情绪的“7C”路径。“7C”是指：“轻视(contempt)”“谨慎(caution)”“自信(confidence)”“深信(conviction)”“安心(complacency)”“关注(concern)”和“投降(capitulation)”。在牛市顶峰和熊市低谷两个市场周期的极端，投资者的心理具有很高的一致性，过度自信在牛市的顶峰达到它的最高点，而在熊市的底端达到最低点。同样，在中国 A 股市场，也存在着类似的路径。

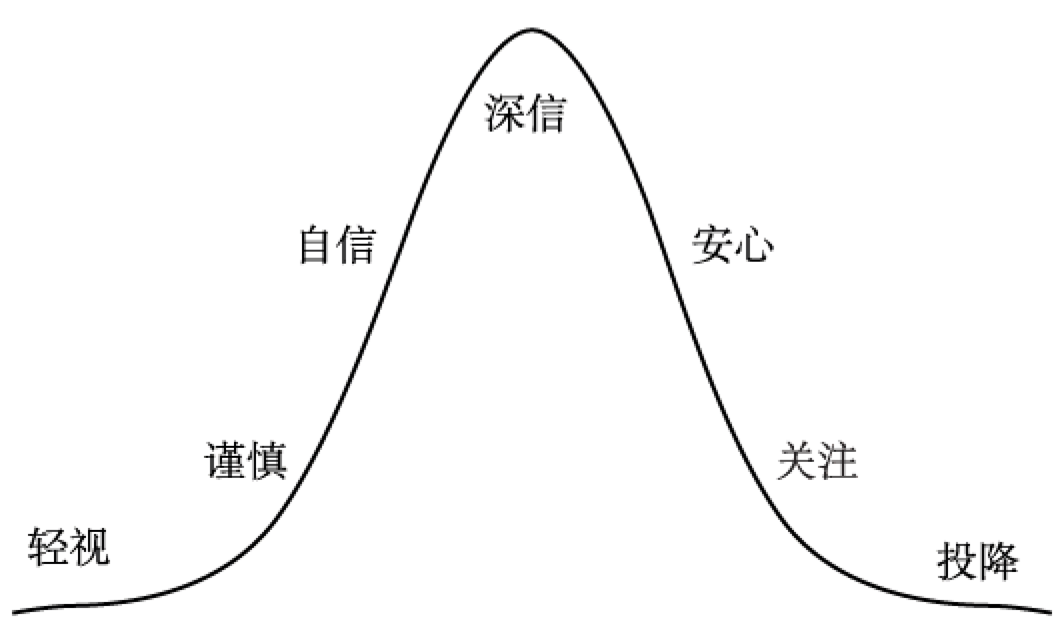


图 2.1　市场情绪周期图

巴菲特有句至理名言：“在大众贪婪时你要缩手，在大众恐惧时你要进取。”中国古人也早有类似的理财思想，司马迁在《史记·货殖列传》中总结了范蠡经商的宗旨，认为“人弃我取，人取我与”是范蠡出奇制胜的重要策略。范蠡主张“夏则资皮，冬则资絺(chī)(细麻布)，旱则资舟，水则资车，以待乏也”，意思就是在夏天做皮草的生意，在冬天做布匹的

生意，在旱灾时做船的生意，发生水灾时做车的生意。这种超前市场的采购没有人竞争，成本比较低，而机遇到来时，收益往往比较高。

近年来行为金融学的研究也认为金融市场行为能够由投资者的情绪加以解释。席勒在《非理性繁荣》中谈到20世纪90年代的股市时认为："投资者的情绪状态是导致牛市最重要的一个因素。"David和Tyler(2003)研究了26个国家的股票交易所所在城市的天气与市场指数日收益率之间的关系，证明阳光充足的晴朗天气，会使人们更加乐观，股票市场的日收益率显著提高。Kamstra和Levi(2002)报告了当交易者的睡眠模式由于夏时制而被打乱时股票市场会下跌。当然，这些关于人们的情绪对股票市场定价影响的定量研究是否是个人行为转化为市场结果的证据，或者说，投资者的情绪变化是否会转化成市场的情绪，进而影响市场，仍然存在一定争议。无论如何，很多专家和学者探索了一个新的研究视角，对投资者，包括散户和机构投资者的心理与行为的研究越来越多，并逐渐构建起行为金融学的心理学框架和基础。

人们在金融市场中的活动包括个体的决策和行为和群体的行为。研究个体决策过程与行为的心理学分支主要有认知心理学和决策心理学；研究群体行为的心理学分支主要是社会心理学。认知心理学研究人的认知过程。决策心理学研究心理现象及其规律在决策过程中的作用和影响机制，包括：①行为决策理论，如最大期望效用理论、主观期望效用理论、前景理论；②影响决策行为的因素，如情绪、个性、动机、态度等对决策行为的影响。社会心理学则研究个体和群体在社会相互作用中的心理与行为的发生及变化规律，其中与金融市场密切相关的内容有"从众、群体影响和偏见"等。

本章第一部分介绍判断与决策中的认知过程。主流经济学把决策者看作是理性的，行为经济学家则认为决策者是"有限理性"的。在人类有限理性的基础上，当人们面临决策时，他们就必须采用简捷的方法(启发式)来配置和使用其资源。第二部分介绍了四种重要的判断启发式：代表性启发式、可得性启发式、锚定和调整启发式以及情感启发式，及其可能导致的偏差。第三部分介绍了框定依赖和框定依赖偏差。

2.1 判断与决策中的认知

1. 认知

"认知"是心理学界普遍使用的一个心理学术语。广义的认知与认识的含义基本相同，指个体通过感觉、知觉、表象、想象、记忆、思维等形式，把握客观事物的性质和规律的认识活动。狭义的认知是指个体获取信息并进行加工、储存和提取的过程。认知是个体重要的心理活动，是人的意识的集中表现。认知对人的情绪、行为具有重要的调节作用。

从广义上说，认知心理学泛指一切以认知过程为对象的心理学研究，包括结构主义认知心理学、心灵主义认知心理学和信息加工认知心理学。狭义的认知心理学专指信息加工认知心理学，是目前西方心理学界通常所指的认知心理学。信息加工认知心理学又称现代认知心理学，是当代西方用信息加工观点和方法研究认知过程的一种新的研究方向，认为认知过程就是信息加工过程，包括感觉输入的变换、简约、加工、储存和实用的全过

程。信息加工认知心理学对认知活动的研究涉及以下四个方面的内容。

1）认知过程

认知过程是个体认知活动的信息加工过程。认知心理学将认知过程看成一个由信息的获得、编码、储存、提取和使用等一系列连续的认知操作阶段组成的按一定程序进行信息加工的系统。

在认知过程中，通过信息的编码，外部客体的特性可以转换为具体形象、语义或命题等形式的信息，再通过储存保留在大脑中。这些具体形象、语义和命题实际就是外部客体的特性在个体心理上的表现形式，是客观现实在大脑中的反映。认知心理学将在大脑中反映客观事物特性的这些具体形象、语义或命题称为外部客体的心理表征，简称表征。通常，"表征"还指将外部客体以一定的形式表现在大脑中的信息加工过程。

2）认知风格

认知风格是个体习惯化的信息加工的方式，又称认知方式。认知风格是个体在长期的认知活动中形成的稳定的心理倾向，表现为对一定的信息加工方式的偏爱。个体常常意识不到自己存在这种偏爱。例如，在获得信息时，有人喜欢从外部环境中寻找，易受外部环境的影响，有人则从认知目标本身中探索，不易受外部环境的影响；在回答问题时，有人倾向于迅速回答，有人则只在有把握时才回答问题；在阅读材料时，有人重视细节，有人则重视整体。

3）认知策略

认知策略是指导认知活动的计划、方案、技巧或窍门。人脑的信息加工能力是有限的，不可能在瞬间进行多种操作，为了顺利地加工大量的信息，人只能按照一定的策略在每一时刻选择特定的信息进行操作，并将整个认知过程的大量操作组织起来。因此，认知策略对认知活动的有效进行是十分重要的。通常人们所说的"如何解决问题""如何保持注意""如何记忆"，指的都是认知策略。

4）元认知

元认知是个体对自己的认知活动的认知。元认知由三种心理成分组成：一是元认知知识，主要包括个体对自己或他人的认知活动的过程、结果等方面的知识；二是元认知体验，指伴随认知活动而产生的认知体验和情感体验；三是元认知监控，指认知主体在认知过程中，以自己的认知活动为对象，进行自觉的监督、控制和调节。

2. 认知过程中的不完美之处

认知过程中的不完美之处表现为有限注意力和认知偏差两个方面。

一是有限注意力。大脑的注意机制类似于一个位于"瓶颈"处的过滤器，把与自己不相干的信息或无用的信息过滤掉，只让有用的信息进入大脑，以减轻大脑的负担。只有重要的信息才会被接收，有用的信息会被保留，没用的信息就会衰减。在当今信息爆炸的社会，注意力的有限性严重地制约了人们对现实世界认识的全面性。如西蒙（Simon，1971）断言，现代社会最基本的稀缺是注意力的稀缺。有关的研究指出人们在一天所遇到的 150 多个广告中只能注意到其中的 1%～7%。

二是认知偏差。人们在认知过程中会尽力寻找捷径。人类是"认知吝啬鬼"（cognitive

misers)，即人们总是在竭力节省认知能量。考虑到有限的信息加工能力，人们总是试图采用把复杂问题简化的策略，常用以下几种方式实现这个目的：

(1) 通过忽略一部分信息以减少认知负担；

(2) 过度使用某些信息以避免寻找更多的信息；

(3) 接受一个不尽完美的选择，并认为这已经足够好了。

认知吝啬鬼策略可能是有效的，因为这样做可以很好地利用有限的认知资源来加工几乎无穷无尽的信息。但这样做也会产生错误和偏差，特别是在选择了错误的策略，或者在匆忙中忽略了重要信息的情况下。

专栏2.1　认知反射测试(CRT)

由普林斯顿大学心理学家夏恩·弗雷德里克(Shane Frederick)在2005年提出一个认知反射测试，评估个体从容而理性地处理信息，而不是急于下结论的能力。测试只有三道题目。

1. 一个球拍和一个球总计1.1美元，球拍比球贵1美元，请问球多少钱？

2. 如果5台机器5分钟造了5个小零件，那么100台机器造100个小零件需要多少分钟？

3. 在一个湖里有一片睡莲莲叶，每天莲叶面积都比前一天多一倍，如果第48天的时候莲叶将把湖面占满，那么请问，第几天的时候莲叶占据湖面一半？

研究发现不少哈佛、耶鲁的高材生都不能完全正确地回答以上三个问题。这表明很多人都是认知的“吝啬鬼”，总是竭力节省认知能量，迅速作出判断，从而产生决策偏差。

(正确答案分别是：0.05美元；5分钟；47天。)

资料来源：FREDERICK S. *Cognitive reflection and decision making*[J]. *Journal of economic perspectives*，2005，19(4)：25-42.

3. 理性、有限理性和生态理性

长期以来，人类理性被看作人和其他动物区别的最重要特征之一，很多年来，经济学中占主导地位的观点是用理性假设解释和预测人的行为，视人为理性决策者，认为人总是在追求个人利益的最大化，从而在有限的环境资源中努力作出最佳的决策，这就是传统经济学奉行的理性人假设。而在心理学家看来，决策过程和问题解决是密切联系的，问题解决总要包括一定的决策，不过问题解决更侧重于产生想法，而决策侧重于从多种想法中加以选择。心理学家对人的决策心理也进行了大量研究，如卡尼曼和特沃斯基提出了前景理论，他们的模型中除了考虑收益(因素1，通常用货币值表示)和概率(因素2，输赢的可能性)因素外，还增加了两个因素：一个是问题表述的语境(因素3)，另一个是人对前三个因素的内部加工(因素4)。

获诺贝尔经济学奖的西蒙则更多用逻辑分析对传统经济学的理论提出质疑，他认为经济学家的决策理论没有考虑到人的认知系统的局限性，为此提出了“有限理性”(bounded rationality)的概念，并指出个体在做决策时远远达不到经济学理论中假定的“最大化”(maximizing)和“最优”(optimize)，而只能达到“满意”(satisficing)。在现实世界中，一个

好的决策不一定是最佳的选择，只要是有效的选择就够了，人的信息加工系统只需要做到令人满意，不必最优化。这个“满意”概念假定个体会选择令自己满意的决策，而不是无限制地搜索所有可能的选择、评估每种选择的概率和效用、计算期望值，然后选择分数最高的项目。

西蒙对决策研究的主要贡献在于：一是强调决策过程中的认知过程，认为决策理论应该尊重人的认知局限性；二是在概率判断研究中强调启发式的应用。一些心理学家受到西蒙的影响，开始致力于研究人在决策中的启发式策略。

德国柏林马克斯·普朗克人类发展研究所心理学家 Gigerenzer 领导的适应行为与认知研究组（Adaptive Behavior and Cognition，ABC）深受西蒙和卡尼曼等人研究的影响，在“有限理性”概念的基础上提出了“生态理性”（ecological rationality）的概念，进一步强调人的认知局限性是具有适应性的，将关于有限理性的研究向前推进了一大步。他们认为人类和动物的理性是有限的（既不是非理性的，又不是纯理性的），但这种有限理性足以使他们在现实环境中作出合理的判断和决策。现实环境并不苛求人类和动物，也就是说并不要求人类和动物时时作出最优化选择与决策，所以任何人都不必为自己理性资源的有限而忐忑不安，相反，那种奢望通过无限理性实现最优化目标的理想主义者反而是不合时宜的。“生态理性”对理性的最低要求是：能够与现实环境（包括自然和社会环境）的要求相匹配，当个体能够做到这一点时，从生态学角度看就已经足够了。

这一观点的一个重要假设是：个体是否理性或其作出的判断和决策是否合理，应该用现实的外在标准来判断，而不是用唯智论者所推崇的不切实际的理想标准来判断。不存在与现实环境相脱离的不着边际的抽象理性，如果有，它对个体适应现实环境也是毫无价值的，只有当与环境的现实要求结合起来考察人类和动物理性时，才能找到评判理性的合理标准，这个标准就是“生态合理性”。

既然人类和动物的理性是有限的，那么当面临上述诸多现实任务时，他们就应该且必须采用简捷而“精明”的方法来配置和使用其资源，因此他们在生态理性的基础上，提出了“简捷启发式”（fast and frugal heuristics）。这种“简捷启发式”借助满足人类有限时间、有限知识和有限的认知能力的简单的心理学原则行事。其主要关注点是人类使用什么样的规则去搜索信息、停止搜索信息以及基于已搜索的信息作出决策，相应地，他们认为人类的决策原则由三部分组成：信息搜索原则、停止信息搜索原则和决策原则。

目前心理学家关于不确定情境下决策研究的一个趋势是继续寻找更多的决策中的启发式策略，寻找这些启发式策略的产生背景和发展轨迹；另一个趋势是越来越多的心理学家关注情绪和情感与决策的关系，而情绪以往是被排除在理性分析以外的，有的心理学家还提出了“情感启发式”（affect heuristics）策略。

总体来说，以往关于人类决策问题的研究，经济学家和心理学家的不同之处在于：经济学家假定环境资源是稀缺的，但却认为人的心智资源是无限的；而心理学家则认为人类的学习、记忆以及决策都受到心理资源的限制，他们致力于人和其他动物适应环境的机制。但是现在经济学家和心理学家关于决策研究的兴趣已经扩展到很多领域，两者关心的领域交叉部分越来越多，如卡尼曼在20世纪90年代研究主观经验和决策的元认知问

题，一些研究者关注个人对自己和他人决策的评价、决策风格的个体差异等问题。

2.2 启发式与偏差

如果将投资者在金融市场上的所有投资活动和各类行为看作一个心理过程与行为结果的话，那么就存在各类认知和行为的偏差。这些偏差有些是系统性的，有些是非系统性的。个体偏差加上金融市场上可能产生的群体偏差或羊群效应(herding effect)，会导致投资组合中的决策偏差，进而导致资产定价的偏差。由资产定价偏差产生的锚定效应(anchoring effect)或框定效应(framing effect)，反过来影响投资者对资产价值的判断，产生进一步的认知与行为偏差，从而形成一种正反馈机制。金融市场中关于认知与行为偏差的研究就是分析投资者各种心理特征，并以此来研究投资者的决策行为及其对资产定价和资产组合问题的影响。

在心理学家看来，决策过程和问题解决是密切联系的，问题解决借助于思维活动过程，侧重于产生想法，决策侧重于从诸多想法中加以选择。人们在解决问题时应用的策略一般有两类：算法(algorithm)和启发式。算法是解决问题的一套规则，它精确地指明解题的步骤。启发式是凭借经验解决问题的方法，是一种思考上的捷径，是解决问题的简单而笼统的规律或者策略，也称为经验法则或者拇指法则(the rule of thumb)。

算法与启发式是两类性质不同的问题解决策略。心理学家一般认为，人类解决问题，特别是复杂问题，主要是应用启发式。卡尼曼和特沃斯基在20世纪70年代早期一系列具有开创性的研究明确了三种重要的判断启发式：代表性启发式(representativeness heuristic)、可得性启发式(availability heuristic)以及锚定和调整(anchoring and adjustment)启发式，此外还有情感启发式，这四种方法既可以得出正确的推理结果，也可能导致错误的推论。错误的推理结果以心理偏差的形式表现出来，这就是所谓的启发式偏差(heuristic bias)，它是指心智正常、受过良好教育的人也会一贯作出错误的判断与决策。

专栏 2.2 在什么情况下，人们最有可能使用启发式，而不是理性思考？

阿伦森(2001)的研究显示至少有四种情况：

(1) 当我们没有时间认真思考某个问题时；

(2) 当我们负载的信息过多，以至于无法充分地对其进行加工时；

(3) 当问题并不十分重要，以至于我们不必太过思考时；

(4) 当我们缺乏作出决策所需的可靠的知识或者信息时，我们更容易使用启发式。

随着互联网和信息技术的发展，人们获取信息的成本大幅下降，获取信息的数量则呈指数级上升趋势。对中国股票市场的大部分投资者而言，他们往往不具备专业的投资知识，却深陷于铺天盖地的股市信息。在这种情况下，投资者很难进行理性思考，只能更多地依靠过去的经验、盲目模仿他人的决策。

心理学家认为人类是“认知吝啬鬼”，即人们总是竭力节省认知能量，试图采用把复杂问题简化的策略。例如，通过忽略一部分信息以减少认知负担，过度使用某些信息以避免

寻找更多的信息，接受一个不尽完美的选择等。认知吝啬鬼策略可能是有效的，因为这样做可以很好地利用有限的认知资源来加工几乎无穷无尽的信息，但这些策略同样会产生错误和偏差。了解认知中固有的局限性和普遍存在的偏差，可以帮助人们更好地作出判断和决策。下面就具体讨论在判断和决策认知过程中的启发式及其偏差。

1. 代表性启发式

代表性启发式是指人们倾向于根据样本是否代表(或类似)总体来判断其出现的概率，即代表性越高的样本，人们判断其出现的概率也越高。例如，人们一般认为从A盒子(70个红球，30个白球)中取出4个白球、2个红球的概率，小于从B盒子(70个白球，30个红球)中取出4个白球、2个红球的概率。这可以看作是代表性启发式的作用：4个白球、2个红球的样本与B盒子中白多红少的构成更类似。

代表性启发式有以下两种情况：一是人们在不确定条件下，会关注一个事物与另一个事物的相似性，以推断第一个事物与第二个事物类似之处。二是人们假定将来的模式会与过去相似并寻求熟悉的模式来作判断，并不考虑这种模式的原因或者模式重复的概率。具体表现在人们会根据事物的一些突出特征对其进行归类，若一个客体或一个人具有的显著特征可以代表所想象的某一范畴的特征，则它易被判断为属于该范畴。认知心理学将这种推理过程称为代表性启发式。

代表性启发式与贝叶斯规则的预测在某些特定场合可能是一致的，从而造成人的概率推理遵循贝叶斯规则的印象。它是人们在处理现实世界问题时所走的捷径，这种推理方法在大多数情况下是有效的。但人们在分析以往经验，寻找规律或结果的概率分布过程中，可能会产生严重的偏差，从而导致判断错误。使用代表性启发式可能产生的偏差有：忽视结果的先验概率，对样本规模不敏感，对偶然性的误解，对均值回归的误解以及有效性幻觉。

1) 忽视结果的先验概率

对代表性没有影响但对概率有很大影响的因素之一就是结果的先验概率(prior probability)，或基础比率(base rate)，引用卡尼曼的两个著名的实验。

实验一：约翰，男，45岁，已婚，有子女；他比较保守，谨慎并且富有进取心；他对社会和政治问题不感兴趣，闲暇时间多用于业余爱好，如做木匠活和猜数字谜语。

假设他来自一个由工程师和律师组成的样本群。实验者分别告诉被试不同的先验概率。一组被试被告知工程师人数为样本的30%，律师为70%。另一组被试被告知工程师人数为样本的70%，律师为30%。询问两组被试约翰更有可能从事哪种职业？

结果表明，两组被试大都认为约翰是工程师，即使在实验者有意提醒他们注意叙述条件的情况下，这种现象仍未改变。这说明，人们只根据描述性语言的代表性进行判断却全然不考虑先验概率的影响。可以注意到这个描述是纯噪声，因为它没有揭示任何与该人是工程师还是律师有关的信息。一般来说，当给出有用的证据时，先验概率会被很合理地使用，当给出没有价值的证据(噪声)时，先验概率却容易被忽视。

实验二："琳达，31岁，单身，性格外向，哲学毕业，在学校期间关心歧视和社会公正问题，参加过反核武器抗议示威活动。"向被试询问琳达更有可能是哪一种人，请被试将下列

叙述按照可能性顺序排列,1代表最可能,8代表最不可能。

(1) 她是一所初级中学的老师。

(2) 她在一家书店工作,并参加瑜伽课程。

(3) 她在女权运动中非常活跃。

(4) 她是精神病治疗的社工。

(5) 她是妇女选举团的成员之一。

(6) 她是银行出纳员。

(7) 她是保险销售员。

(8) 她是银行出纳员并且热衷于女权运动。

从实验人数来看,80%~90%的人认为第(6)项比第(8)项更有可能。

更为有趣的是,实验者按照被试的概率知识将被试分成三组。

第一组:知识水平一般的被试(来自UBC和Stanford的没有概率或者统计学背景的本科生)。

第二组:知识水平中等程度的被试(来自Stanford心理学、教育学以及医学的研究生,曾学过概率与统计的课程)。

第三组:知识结构完备的、老练的被试(Stanford商学院决策科学项目的研究生,学习了概率和统计的高级课程)。

这三组的结果几乎是一样的,相似性排序与可能性排序完美地一致。虽然选项(8)出现的概率要比选项(6)出现的概率小得多,琳达是一个女权主义的银行出纳员的可能性小于她是一个银行出纳员的概率,不过人们似乎认为选项(8)是对琳达更自然的描述,更像她的代表性特征,他们使用可代表性启发式,有关琳达与一个女权主义者形象契合的描述成为联系这一选择的加权因素。

概率论中的结合率表明,某事物既属于范畴A又属于范畴B的概率不会大于且往往小于它单独的属于A或者B的概率,即$P(AB)\leqslant P(A)$或者$P(B)$,而在实际认知决策的时候,人们往往遵循关联效应(认为关联的事件具有更高的可能性),表现出了与概率论的结合率相反的情况。

2) 对样本规模不敏感

代表性还会导致另一个偏差,即忽略样本大小。在分析事件特征或规律时,人们往往不能正确理解统计样本大小的意义。人们不考虑样本的大小,以为小样本和大样本一样都具有代表性。例如,认为"扔6次硬币出现3次头像"与"扔1 000次硬币出现500次头像"一样具有代表性。实际上对总体进行统计的结果才是真正的结果,样本的数量越接近真实的数量,统计的结果也就越可信;样本越小,与真实数量相差越大,统计的结果越不能反映真实的结果。

代表性启发式的推理过程,要求人们首先对同类事件以往所出现的各种结果进行统计分析,得到结果的概率分布,从而找出发生概率最大的结果,或最可能发生的结果。因此,人们必须考察所有同类事件这个总体,或者考察尽量多同类事件(大样本),在此基础上应用代表性启发式,才有可能得到正确的推理。但是在现实中,人们往往趋向于在很少

的数据基础上很快地得出结论，忽略样本规模的大小，或者对样本规模不敏感。人们都有这样的经验：如果一个人连续几次都预测正确，人们就会十分相信他的判断力，认为他下次的预测仍会是正确的。再如当一个球星在一场球赛中踢进3个球时，球迷会确信该明星处于顶峰时期，即使没有其他数据可以证明该球星是个能手。这种认为小样本可以反映总体现象的观点被称为"小数定律"(law of small numbers)，即人们错误地认为小样本和大样本有相同类型的概率分布。

例如某镇有两家医院：大医院里每日出生45名婴儿，小医院里每日出生15名婴儿，婴儿50%为男孩，但具体的比例每天都在改变。1年来，两家医院都把出生的男婴数目超过60%的天数做了记录。那么哪家医院记录的天数更多些？实验结果：有21人认为大医院记录的天数多；有21人认为小医院记录的天数多；有53人认为两者记录的天数大约相同。

3）对偶然性的误解

人们认为，一个由随机过程产生的事件结果可以代表该随机过程最本质的特征。

当人们知道了某事件的无偏(客观)概率的情况下，往往会对已经发生的小样本事件进行错误的估计，从而影响对未发生事件的概率估计。例如在抛硬币时(H表示头像，T表示反面)，如果硬币是公正的，实验者在一连串HHH后会期待下一个为T。人们会错误地认为HTHTHT这个结果比HHHTTT这个不随机出现的结果，或者比HHHHTH这个不能代表硬币公正性的结果更有可能出现。这是由小数定律引起的"局部代表性"(local representativeness)。

局部代表性观念的一个结果是"赌徒谬误"。所谓赌徒谬误，就是指对于那些具有确定概率的机会，人们会错误地受到当前经历的影响而给予错误的判断。人们相信，当某一事件最近发生过后，该事件未来发生的概率会降低，虽然事实上其客观概率是确定的。人们都知道投币得到正反面的概率相等，因此当连续多次投硬币都得到正面的情况下，人们总倾向于认为下一次出现反面的概率很大。

专栏2.3　抛硬币中的赌徒谬误

如果抛一枚相同的硬币8次都是头像，那么第9次抛得反面的可能性是多大呢？

人们会认为头像出现的次数已经够多了，第9次抛得反面的概率应该大于50%；当然，也有人认为前面总是出现正面，意味着正面出现的机会更大。

两种解释都不对，实际上第9次抛掷出现正面或反面的概率仍然是50%。

类似的实验：如果女孩与男孩出生的可能性一样(B代表男孩，G代表女孩)，那么实验者会期待在一系列GGG后出现B，人们会认为BGGBBG的结果比BBBBBB更有可能出现。

4）对均值回归的误解

在随机过程中，变量可能按随机游走分布，或者符合均值回归的趋势，均值回归变量遵守自我矫正过程。人们往往作出"非回归预测"(nonregressive predictions)，也就是说他们用线性的方式对问题进行预测与推断，而没有考虑到，现实中由于种种因素的影响，事情的发展趋势往往存在回归的倾向。例如，对于一些如通货膨胀率的变量来说，是很难判断它们是否是固定的；而其他一些如公司业绩变量则很容易判断，均值回归认为公司未来

的业绩会接近它们的历史均值。

特沃斯基等(1982)给出了一个例子,在一个关于飞行训练的讨论中,有经验的教官注意到飞行员在一次平稳的着陆受到表扬之后,通常在下一次着陆时则表现很差,而对一次不理想的着陆进行严厉的批评之后,下一次的试飞就会有巨大的进步。教官错误地得出结论,认为言辞上的奖励对学习不利,而言辞上的批评则是对学习有利的。这一结论是没有根据的,因为短期业绩是个均值回归的变量。随机性表现在"退步"出现在好的着陆之后,而"进步"出现在差的着陆之后,但是人们直觉上都希望后续业绩是先前业绩的代表,所以忽视了对回归到平均值的趋势。

5）有效性幻觉

在预测结果与输入信息之间良好吻合的基础上形成的没有根据的自信,被称为"有效性幻觉"(illusion of validity)。当输入变量非常多并且相互关联时,属于高度一致的输入模式。某一输入模式的内部一致性是人们对以这些输入为基础作出判断的信心的主要决定因素,这样,人们对这种多余的、相互关联的、可是并不能提高预测准确性的变量的预测会变得很有信心,然而这种信心很有可能仅仅是一种有效性幻觉而已。例如,人们在预测某一学生的分数全部是 B 时比该学生第一学年的分数包含很多个 A 或 C 时更自信。

人们经常通过选择最能代表输入(如对某个人的描述)的结果(如某一职业)来进行预测。如当人们要求被试利用某个人的描述资料去预测其职业时,被试常常选择最能代表个人描述特质的职业来做预测,而且当描述资料和所选的职业间相似的程度越高,被试对自己所做的判断越有信心。人们对他们预测的信心主要依赖于代表性的程度,而对影响预测准确性的因素(如信息不充分、不可靠或已过时等)考虑很少或甚至不考虑。

2. 可得性启发式

可得性启发式是指人们倾向于根据客体或事件在知觉或记忆中的可得性程度来评估其出现的相对频率,容易察觉到的或回想起的客体或事件被判定为更常出现。它常用于判断某事件发生频率或者概率的情况,这时人们往往会利用记忆中最容易提取的信息进行主观估计,这在事件的可得性与其客观概率高度相关时是非常有用的,但是由于人们对频率或者概率的估计极大地受对事件熟悉程度与显著性等可获取性因素的影响,并受搜索方式的有效性和事件可想象的难易程度的影响而造成认知偏差。

投资者在实际投资时其决策与该问题的资料信息是否充分、是否容易获取有关。很多时候,人们只是简单地根据信息获取的难易程度来确定事件发生的可能性。例如,相对于新入市者而言,体验过"跳水"的老投资者在面临股市下跌风险时,更容易高估股票价格直线下降的可能性。

影响可得性启发式偏差的因素有:①时间的接近性。某件事越是最近发生过,越容易被经常想起,越会被判定发生的概率高。例如,如果 1 个月内连续两次飞机失事,很多人会因此高估飞机失事的概率,从而不愿意乘坐飞机。②新异性。事件越是新鲜和与众不同,人们对其关注的程度越高,从而可能会高估其结果(如我国普遍存在的"消息市""政策市")。③表现的生动性。越是表现生动的事物,人们的记忆越深刻。生动的信息对决策者的影响更大。④情绪一致性。人们对某事件的记忆往往会和该事件引起的情绪结合起

来。以后当该情绪再次出现时，也就容易回忆起该事件，高估该类事件发生的概率。⑤想象性。越是容易被想象的事物，人们容易想象得越多，从而越会认为自己的想法正确，这是一个不断加强的陷阱。

现实金融市场中有许多这方面的例子，席勒(2000)通过调查发现，由于20世纪90年代后的股市繁荣伴随着网络的迅速发展，网络使用者倾向于将股市繁荣归功于网络的发展。由于网络发展给人的印象比较深刻，相对于其他的事情，这些投资者认为网络在这一轮牛市行情中起着更重要的作用。

可得性启发式导致的偏差主要体现为以下四种。

1）由于事件的可获取性而导致的偏差

人们都有过类似经验，当被问及某种疾病的发生概率时，人们往往根据身边熟悉的人得此病的情况进行推断；又如，若被问道：每年因飞机失事和被毒蛇咬伤而死亡的概率哪个大？这时人们往往更容易选择前者，虽然也许事实上二者发生的概率相同，但由于舆论传媒的导向作用，使得人们对前种情况更为熟悉和敏感。在影响事件可获取性的因素中，除了是否熟悉以外，还有其他一些因素，如显著性。例如，看到房子被烧对这种意外的主观概率判断的影响要比读报纸了解到一场火灾对主观概率判断的影响更大。

2）由于搜索方式的有效性而导致的偏差

能够很好说明搜索方式的有效性的一个例子是：以r(如ride)开头的单词，还是第三个字母是r(如circle)的单词多？这时也许会马上很容易地从记忆中提取一些单词，如red、roof等一系列以r开头的单词，却很难短时间内提取第三个字母是r的单词，于是就作出判断，以r开头的单词多。但实际上，仅仅是因为对第二种情况不够熟悉，认为第三个字母这一提取线索是无效的，真正的情况恰好与选择相反。

3）“意象”偏差

人类对某事物心有所感，便将之寄托给一个所选定的意象，使之融入自己的某种感情色彩，并制造出一个特定想象情境。于是，人们根据经验对某一现象进行情境推断，但是实际情况可能与之有误，意象偏差就形成了。

一旦人们对某事件没有经验的时候，人们往往会根据该事件可想象的难易程度来判定它的发生概率。例如，从一个81人组成的样本中抽取2人组成委员会和从一个同样大的样本中抽取79人组成委员会两种情况的组织方式各有多少种？分别让两组被试对以上两个情境之一进行直觉判断，发现在第一种情况下，被试估计的数量要远大于第二种情况。而实际上，由任意两人组成一种委员会形式的同时，样本中余下的79人就相应地形成一种组织方式，因此两种情况下的答案应是相同的。之所以会发生这种偏差，就是因为两人的组成情况更容易进行想象。再如，对一次冒险性探险的风险进行评估，是通过探险的装备无法解决的偶然性来评估的，如果有些困难很容易被想象到，那么探险看起来是十分危险的，即使这些容易想象的危险并不具有很大的可靠性，相反，如果一些潜在的危险不容易被人们想象到，那么，风险可能总体上会被低估。

4）虚幻的相互作用

人们在日常判断和决策中，总是运用与事件有联系的虚幻现象来对事件进行判断，从

而可能产生偏差。当决策者根据两个事件在脑海中相关联的程度，来判断两个事件共同发生的次数或概率时，就会产生偏差，主要因为决策者无法了解事件间真正的关系，或者因为先入为主的观念影响了决策者的判断。此时，若决策者认为两个事件相关联的程度很高，便会判断两事件共同发生的概率很高，但这种判断方式往往高估了这一概率。

在一个实验室中，被试被要求对一些假想精神病人信息作出判断，病人的数据包括临床诊断和这个病人的画像，判断估计每一诊断（如多疑症或妄想症）和他画中所描述的各种特征（如奇怪的眼睛）相符的概率。受调查者明显高估了自然联系共同发生的概率，如多疑症和奇怪的眼睛。即使决策者已获知相反的证据，仍会产生幻觉上的相关，即先入为主的观念很难加以改变。

苏联社会心理学家包达列夫做过这样的实验，将一个人的照片分别给两组被试看，照片的特征是眼睛深凹、下巴外翘。向两组被试分别介绍情况，给甲组介绍情况时说“此人是个罪犯”，给乙组介绍情况时说“此人是位著名学者”，请两组被试分别对此人的照片特征进行评价。

评价的结果，甲组被试认为：此人眼睛深凹表明他凶狠、狡猾，下巴外翘反映其顽固不化的性格；乙组被试认为：此人眼睛深凹，表明他具有深邃的思想，下巴外翘反映他具有探索真理的顽强精神。

3. 锚定与调整启发式

在判断过程中，首先人们通常会根据最初得到的信息设定一个判断的初始值（锚点，参考点）；然后根据进一步获得的信息进行调整，形成比较理想的判断。

所谓锚定效应，就是指当人们需要对某个事件做定量估测时，会将某些特定数值作为起始值，起始值像锚一样制约着估测值，人们通常以一个初始值为开端进行估计和调整，以获得问题的答案。锚定效应的起始点（starting point）可以是事件本身提供的，也可以是在估计过程中局部发生的，但是无论哪一种情况，起始点都会对估计值产生决定性影响。调整策略是指以起始点的信息为参照来调整对事件的估计，调整通常是不充分的，不同的起始点会产生不同的估计，这就容易偏离其初始值。锚定与调整启发式导致的偏差主要有调整不足引起的偏差，如“价格黏性”（sticky prices），“货币幻觉”；对联合和分离事件评估时的偏差，倾向于高估复杂系统成功的概率和低估其失败的概率。

锚定效应引发的偏差主要有两种。

1）调整不足引起的偏差

调整不足亦称反应不足，系指人们对于信息反应不准确的一种形式，亦即人们本身即存有一种惰性，不愿意改变原有的信念，即使有了新的信息，人们的反应还是不足或调整不充分。特别是，当人们在对后面的事件进行判断时，对前面类似现象的印象制约着后面的判断范围，导致了调整不足。为了验证人们是否具有调整不足所引起的偏差行为，在一个幸运轮实验中，特沃斯基与卡尼曼曾要求被试估计各种不同的数值，并用百分比加以表达。就每一数值，研究者先在被试面前，以幸运轮摇出一个数字作为起点，首先让被试回答这个数字高于或低于他们估计的数值，然后再让其说出其心中真正的估计值，结果发现，这个没有特殊意义的起点对被试的判断有显著的影响，被试给出的估计值受到了幸运

轮产生的随机数字的深刻影响；且即使对于正确的估计给予报偿，亦并未降低定锚效果。例如请被试估计联合国中有多少个非洲国家，当幸运轮停在 10 处，被试回答的非洲国家数量平均数是 25，但是当幸运轮停在 65 处时，平均值就变成了 45。有趣的是，被试都清楚轮盘产生的数字是随机的，并且也不会对被试产生任何情绪的影响。

金融市场中常见的对价格的锚定以及反应不足等现象和锚定与调整启发式有着密切的关系。Cutler，Poterba 和 Summers(1989)发现，当重要消息发生时，股票价格只有少许变动，随后在没有其他什么重大信息透露时发生巨幅变动。再如，大多数资产价格泡沫具有的典型特征是，在最后一个阶段到来前，价格和增值效应通常都会延续相当长时间，这使得投资者改变了预期，认为高价格是合理的。

研究表明，并非所有的数字都会产生锚定效应，产生锚定效应的条件有两个：第一，被试对锚值充分注意；第二，锚值与目标值相兼容，如单位、维度一致。

专栏 2.4　两道计算题

特沃斯基与卡尼曼对两组被试分别提出下列两个问题。

(1) 8×7×6×5×4×3×2×1＝?；(2)1×2×3×4×5×6×7×8＝?。

要求被试在 5 秒内估计出其乘积。

限定这么短的时间是为了不让被试做完整的计算，结果发现，被试对第一道题估计值的中数是 2 250，对第二道题估计值的中数是 512，两者的差别很大，并都远远小于正确答案 40 320。可以设想，被试在对问题做了最初的几步运算以后，产生“锚定效应”，就以获得的初步结果为参照来调整对整个乘积的估计，因为调整是不充分的，因此导致估计不足；但是由于两道题的乘数数字排列不同，第一道题的最初几步的运算结果大于第二道的，因而对其乘积的估计值也较大。

锚定现象和宏观经济学家所谈论的“价格黏性”[①]有关。价格黏性指价格不能随着总需求的变动而迅速变化。工资黏性指工资不能迅速调整。商品的价值越模糊，参考过去的价格就可能越重要，锚定效应就可能是更重要的价格决定因素。只要过去的价格被用作对新价格的建议，新价格就趋于接近过去的价格。

锚定调整偏差在证券市场往往表现为对股票价格的锚定。在判断股票的价格水平时，最可能的锚定数字是记忆中离现在最近的价格，因此使股价日复一日地趋同。过去的股价可能会成为股价发展势头逆转的原因之一。另外，还有股价指数在最近达到顶峰和最近的整体水平，投资者对这些数字的锚定可以用来解释其非同寻常的行为。对单个股票而言，价格的变化也会受到其他股票价格变化的锚定，市盈率也会受其他公司市盈率的锚定。这种联系有助于解释为什么不同的股票会一起涨跌，可以解释为什么不同行业但总部设在统一国家的企业，比同一行业但总部设在不同国家的企业有着更加相似的股价变动趋势。证券市场中出现的这些非正常现象都可以用锚定作出解释。

锚定还导致“货币幻觉”(money illusion)。“货币幻觉”一词是美国经济学家欧文・费

① “价格黏性和货币幻觉”参考：饶育蕾，刘达峰．行为金融学[M]．上海：上海财经大学出版社，2003.

雪(Fisher)于1928年提出来的,它是指人们通常倾向于对货币的名义价值作出反应,而忽视其实际购买力变化的一种心理错觉。凯恩斯认为人们对货币幻觉的免疫是脆弱的,所以他在解释收入分配时,假设工人不会谈判工资增长以抵消通货膨胀。卡尼曼等(1986)通过调查发现,人们在评估工资、价格水平等财富数字时,对名义金额的关注超过了对实际金额的关注。例如,在名义工资上涨5%,而同期通胀率为12%的情况下,人们的不适感要小于工资水平下降7%,而通胀率为零的情况。

特沃斯基等(1997)通过实验表明,人们倾向于依据问题的给出是以名义变量的形式还是以真实数量的形式而对同一问题给出不同的答案。在问题中所给的数量也像锚定一样产生作用。货币幻觉会影响人们在认知上、情绪上处理通货膨胀的方法。例如,一年前,安妮和芭芭拉毕业于同一所大学。毕业后,两人在公司从事相似的工作,安妮的起薪是每年30 000美元,在工作的第一年中,没有通货膨胀,第二年,安妮得到2%(600美元)的加薪。芭芭拉的起薪也是每年30 000美元,在工作的第一年中,有4%的通货膨胀率,第二年,芭芭拉得到5%(1 500美元)的加薪。那么当她们进入第二年的工作中时,谁的经济情况更好?你认为谁最开心?如果两个人都收到另一家公司的工作邀请,你认为谁最可能为了另一份工作而辞去现在的工作?

大多数人指出是安妮更好,芭芭拉最开心。并且,安妮更可能为另一份工作而辞职。现在就有点迷惑了,如果安妮更好,为什么她不开心而且还更可能为了另一份工作而辞职呢?特沃斯基等认为,虽然人们能指出怎么为通货膨胀做调整,但这不是他们通常的思维方法,通常的方法是根据账面价值来考虑,所以账面价值驱动着人们的情感反应。

专栏2.5　股权分置改革中的锚定效应

2005年起,我国股票市场开始进行股权分置改革,此次改革的基本方法是采用非流通股东向流通股东支付"对价"的方式来换取非流通股的上市流通权,以实现"同股同权、同股同价",消除非流通股与流通股的流通制度差异。首批成功实施股改的3家试点公司,非流通股东支付给流通股东的对价分别为10送2.5,10送3和10送3.972,平均为10送3.157。一个有意思的现象是,在此之后不论上市公司的特征千差万别,股改公司的对价大部分都集中在"10送3"水平。其中一个可能的原因是,第一批试点公司的对价方案成为市场的"锚",其他公司均以此为参照,并没有根据自身的情况进行充分调整。

资料来源:许年行,吴世农. 我国上市公司股权分置改革中的锚定效应研究[J]. 经济研究,2007(1):114-125.

2) 对联合事件和分离事件的认知偏差

在一个实验中,被试获得一个机会对两个事件中的一个下赌注,这一实验用到了以下三种类型的事件。

简单事件:从一个装有50%红球和50%白球的箱中抽到一个红球。

联合事件:从一个装有90%红球和10%白球的箱中连续7次抽到红球。

分离事件:从一个由10%红球和90%白球的箱中连续7次取球且至少有一次抽到红球。

先根据统计学原理分别计算以上三种情况各自抽中红球的概率:$P_1=0.5$;$P_2\approx0.48$;$P_3\approx0.52$,因此无偏差的主观估计值应该是$P_3>P_1>P_2$,人们应该偏好分离事件。但事

实上，绝大多数被试喜欢联合事件而最不喜欢选择分离事件。被试在对简单事件和联合事件下赌注时，绝大部分喜欢对联合事件下赌注，而不对简单事件下赌注；在对简单事件和分离事件下赌注时，则喜欢对简单事件下赌注，而不对分离事件下赌注。这种选择模式说明人们倾向于高估联合事件的概率，并低估分离事件的概率。

对联合事件和分离事件评估的偏差普遍存在于按计划进行的工作中。当决策者在评估一个计划能否成功时，即使每一步骤成功的概率都很高，但是由于所必须经过的步骤很多，而使得整体成功的概率较低，如果决策者使用锚定和调整经验法则，便会高估整体成功的概率。相反，在评估一个复杂制度的风险时，即使每一要素发生故障的可能性相当低，然而，由于所需考虑的要素很多，使得整体失败的概率较高，此时，如果使用锚定和调整经验法则，便会低估风险而产生偏差。

在现实中，一项工作的成功完成，如一个新产品的推出，是由很多环节和部件所组成的，每一个环节和部件都有成功与失败的可能性，即使每个环节成功的概率很高，但如果事件数量很大的话，成功的总概率仍会很小。高估联合事件概率的一般趋势，导致人们在估计某一计划将成功或某一项目将按时完成时过于乐观。

4. 情感启发式

“情感启发式”是指依赖于直觉和本能对不确定性事件进行判断与决策的倾向。情感的意义在于运用经验、直觉和本能节省决策的信息费用，当用严格推理来处理信息的费用极其高昂的时候，根据情感作出反应就是一种经济的决策方式。对刺激物的情绪反应常常比认知评估来得既快又及时，这种即时的情绪反应为决策者的行为选择提供了线索和依据，使决策者的快速行动成为可能。情感启发式由于固有的心理缺陷会导致决策者产生偏差。情感启发式是一种进行评价和判断的线索，因为个人的决策行为受自身经验、情感和立场的影响，往往带有明显的选择性特征，如所谓“爱屋及乌”。

心理学研究表明：人们在认知过程中会给不同物体、概念、形象贴上偏好或情感的标签，这些标签会大大影响他们判断的客观性，因为很难剥离出潜在情绪对现实判断目标的影响。同时发现，人们频繁用情感启发式对收益和风险进行决策，今天或者明天的心情和情绪都会影响到投资者对风险的认知与选择。

2.3 框定依赖与框定依赖偏差

1. 框定与框定依赖

根据社会学家欧文·戈夫曼(Erving Goffman)对“框定”的定义，框定是人们解释外在客观世界的心理模式，用来作为了解、指认以及界定行事经验的基础。所有对于现实生活经验的归纳、结构与阐述都依赖一定的框定；框定能确定、理解、归纳、指称时间和信息，同时人们依赖主观认知中的框定来组织经验、调整行动。

在人们形成认知时，人们所面对的背景、对事物的描述和表现方式都会影响到人们对事物的认知与判断。框定依赖，就是指人们会受问题出现的情境或问题表达方式的引导，而对同一事物表现出不同的判断或偏好，从而作出不同的选择。

期望效用理论中的恒定性假设，要求对事件描述的变化不会引起偏好顺序的改变。但是卡尼曼与特沃斯基却发现人们对一个备择方案或风险方案的二择一决策，会受到语言表述方式的影响，他们将这种由于描述方式的改变而导致选择偏好发生改变的现象称为框定效应。这里把对问题的构造或表述方式称为框定，框定不同导致选择不同，即对同一个问题的两种在逻辑意义上相似的说法却导致了不同的决策判断，或者说人们的判断依赖于描述事物的方式，这就是框定依赖(framing dependence)。具体来讲，框定包括：不同方案的比较、事情发生前人们的想法、问题的表述方式或者信息的呈现顺序和方式。

以下是卡尼曼和特沃斯基有关框定依赖的经典实验之一：生存框架(Survival Frame)和死亡框架(Mortality Frame)。

实验设计：假定在一种流行疾病的侵袭下，将有 600 人死亡，请被试(N=152)在两种方案中做出选择。

采用方案 A，将有 200 人获救；

采用方案 B，600 人全部获救的可能性为 1/3，而全部死亡的可能性为 2/3。

试问应该采用哪种方案？

实验结果：72%的人选择方案 A。

然后，对第二组被试(N=155)叙述同样的情景，同时将解决方案改为 C 和 D。

采用方案 C，将会有 400 人死亡；

采用方案 D，则无人死亡的概率为 1/3；600 人全部死亡的概率为 2/3。

请被试选择采用哪种方案。

实验结果：78%的人选择了方案 D。

实验结果分析：事实上，方案 A 与方案 C、方案 B 与方案 D 分别是等值的。但是绝大多数人作出了相反的选择。之所以会出现以上结果，可能是由于实验提问方式的影响，使得第一组被试考虑的是将获救的人数，而第二组被试考虑到了会死亡的人数。因此，第一种情境下人们不愿冒会使更多人死亡的风险；第二种情境下人们则倾向于冒风险救活更多的人。再进一步考察，我们会发现，在不同情境下，人们分别表现出对损失(更多的人死亡)的回避和对利益(更多的人获救)的偏好。

著名的 Müller-Lyer 错觉效应也可以让人们更加直观地理解框定依赖的作用，如图 2.2 所示。图 2.2(a)所示为 Müller-Lyer 错觉，从视觉上来看，似乎上面的直线要比下面的直线长，尽管上面的直线实际上要比下面的直线短；但若将其放在一个长方形的框定中，如图 2.2(b)所示，显然，下面的直线要长于上面的直线，这个长方形框定消除了图 2.2(a)中的错觉。由此可见，对事物本质的判断与认知受到问题表征框定和决策者经历的限制。

2. 框定依赖偏差

在判断和决策过程中，人们会受到信息背景的干扰而产生偏差，形成错误的判断。由于框定依赖而产生偏差的几个主要例子是对比效应(contrast effects)、首因效应、近因效应(recency effect)、晕轮效应(halo effect)和稀释效应(dilution effect)等。

1) 对比效应

在认知心理学中，人们把某一特定感受器因同时或先后受到性质不同或相反的刺激

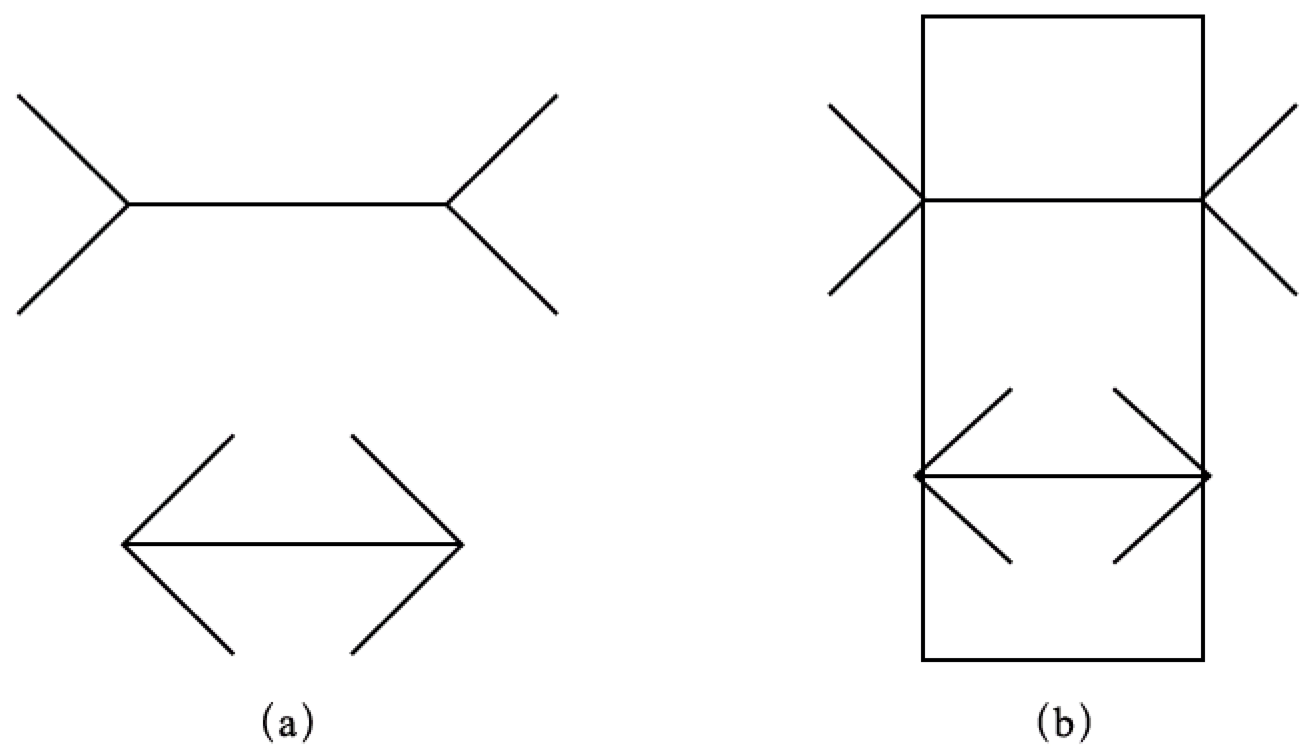


图 2.2 Müller-Lyer 幻觉图形

物的作用,引起感受性发生变化的现象,叫作对比效应,即事物与背景之间的对比越明显,事物越容易被感知。

Plous(1993)做了一个简单的实验,他用 3 个大碗,第一碗盛热水,第二碗盛温水,第三碗盛冰水。被试把一只手浸入热水中,另一只手浸入冰水中,要浸入 30 秒。等手已经适应了水温,把在热水中的手浸入温水中,5 秒后,再把冰水中的手也浸入温水中。

如果你和大多数人一样的话,就会有奇怪的感觉。先前浸在热水中的手会告诉你这碗水是冰的,而先前浸在冰水中的手会告诉你这碗水是热的。事实上,如果让另一个人来做这个实验,并且不要告诉他那碗是温水,他可能就不能辨别出那碗水的温度是多少。每只手都呈现出对比效应。早期的许多心理学研究涉及了诸如温度识别、颜色识别和重量识别等知觉判断。因此,对比效应是最早在实验室里被可靠证明的心理现象之一。

对比效应的研究告诉我们,对比的选择会产生截然不同的效果,根据前后不同的情景,可能让事物或方案看起来更好或更坏。

2) 首因效应

首因,是指首次认知客体而在人们脑海中留下的"第一印象"。首因效应是指个体在社会认知过程中,通过最先输入的信息对客体以后的认知产生的影响作用,也称为"第一印象"作用,或"先入为主"效应。实验心理学研究表明,外界信息输入大脑时的顺序,在决定认知效果的作用上是不容忽视的,最先输入的信息作用最大,最后输入的信息也起较大作用,大脑处理信息的这种特点是形成首因效应的内在原因。例如,上市公司年报及招股说明书通常会在陈述上做文章,会先花费大量笔墨和篇幅来描述与论证自己发展的美好前景,而仅仅花很少的笔墨粗略地描述不利的信息。

美国心理学家卢钦斯(A. Ladins,1957)用编撰的两段文字作为实验材料研究了首因效应现象。他编撰的文字材料主要是描写一个名叫吉姆的男孩的生活片段,第一段文字将吉姆描写成热情并外向的人;另一段文字则相反,把他描写成冷淡而内向的人。在实验中,卢钦斯把两段文字加以组合:第一组,描写吉姆热情外向的文字先出现,冷淡内向的文字后出现;第二组,描写吉姆冷淡内向的文字先出现,热情外向的文字后出现;第三组,只显示描写吉姆热情外向的文字;第四组,只显示描写吉姆冷淡内向的文字。

卢钦斯让四组被试分别阅读一组文字材料，然后回答一个问题"吉姆是一个什么样的人？"结果发现：第一组被试中有78%的人认为吉姆是友好的；第二组中只有18%的被试认为吉姆是友好的；第三组中认为吉姆是友好的被试有95%；第四组只有3%的被试认为吉姆是友好的。

这项研究结果证明，信息呈现的顺序会对社会认知产生影响，先呈现的信息比后呈现的信息有更大的影响作用。但是，卢钦斯进一步的研究发现，如果在两段文字之间插入某些其他活动，如做数学题、听故事等，则大部分被试会根据活动以后得到的信息对吉姆进行判断，也就是说，最近获得的信息对他们的社会知觉起到了更大的影响作用，这个现象叫作近因效应。

专栏2.6　首因效应小故事

一个新闻系的毕业生正急于寻找工作。一天，他到某报社对总编说："你们需要一个编辑吗？""不需要！""那么记者呢？""不需要！""那么排字工人、校对呢？""不，我们现在什么空缺也没有了。""那么，你们一定需要这个东西。"说着他从公文包中拿出一块精致的小牌子，上面写着"额满，暂不雇佣"。总编看了看牌子，微笑着点了点头，说："如果你愿意，可以到我们广告部工作。"这个大学生通过自己制作的牌子表达了自己的机智和乐观，给总编留下了美好的"第一印象"，引起其极大的兴趣，从而为自己赢得了一份满意的工作。这种"第一印象"的微妙作用，在心理学上称为首因效应。

3）近因效应

近因效应指在总体印象形成过程中，新近获得的信息比原来获得的信息影响更大的现象。在人的知觉中，如果前后两次所得到的信息不同，但中间有一个无关工作把它们分隔开，那么后面的信息在形成总印象中起到的作用更大，这种现象是由于近因效应的作用，前后两次信息之间的间隔时间越长，近因效应越明显，原因在于前面的信息在记忆中逐渐模糊，从而使近期信息在短时记忆中更为突出。研究发现，近因效应一般不如首因效应明显和普遍。

首因效应和近因效应各自可能出现的情况为：若信息是连续呈现出来的，但判断或决策是在一段时间后才作出，则此时首因效应是主要的；若信息的呈现之间有时间间隔，但判断或决策是紧接着作出的，则此时近因效应是主要的。

4）晕轮效应

晕轮效应又称"光环效应"或"成见效应"，常表现在一个人对另一个人或事物的某种印象决定了对他的总体看法，而看不准对方的真实品质，形成一种好的或坏的"成见"（错觉）。

晕轮效应最早是由美国著名心理学家爱德华·桑戴克（Edward Thorndike）于20世纪20年代提出的。他认为，人们对他人的认知和判断往往只从局部出发，扩散而得出整体印象，也即常常以偏概全。一个人如果被标明是好的，他就会被一种积极肯定的光环笼罩，并被赋予一切都好的品质；如果一个人被标明是坏的，他就被一种消极否定的光环所笼罩，并被认为具有各种坏品质。

日常生活中，晕轮效应往往是悄悄地却又强有力地影响着对人的知觉和评价。晕轮效应的极端化就是所谓“爱屋及乌”。晕轮效应给人们认知带来的消极影响的直接结果就是偏见，偏见是以有限的或不正确的信息来源为基础的，一个走进晕轮效应迷宫的人，势必会产生偏见。由于晕轮效应的存在，投资者往往会高估或低估股票在某些方面的表现。

专栏 2.7　晕轮效应的实验

美国心理学家凯利（Harold Kelley，1950）以麻省理工学院的两个班级的学生分别做了一个实验。上课之前，实验者向学生宣布，临时请一位研究生来代课，并告知学生有关这位研究生的一些情况，其中，向一个班的学生介绍这位研究生具有热情、勤奋、务实、果断等项品质，向另一个班的学生介绍的信息除了将“热情”换成“冷漠”之外，其余各项都相同。而学生们并不知道两种介绍间的差别。结果：下课之后，前一班的学生与研究生一见如故，亲密攀谈；另一个班的学生对他却敬而远之、冷淡回避。可见，仅介绍中的一词之别，竟会影响到整体的印象，学生们戴着这种有色眼镜去观察代课者，而这位研究生就被罩上了不同色彩的晕轮。

5）稀释效应

“稀释效应”是指中性和非相关信息容易削弱人们对问题实质的判断。稀释效应可以用一个实验来加以描述。在实验中，被试得到了 A、B 两个信息，要求判断提姆更勤奋，还是汤姆更勤奋。

A：提姆平均每个星期要花 31 个小时的课外时间学习。

B：汤姆有一个调皮可爱的弟弟和两个妹妹，他们相处融洽，他每隔一周去看望一次爷爷奶奶，每隔两个月打一次台球，他花在课外学习的时间是 31 个小时。

在上述描述中，提姆和汤姆的勤奋程度是一致的，都是花 31 个小时进行学习。然而，实验发现人们会认为提姆更勤奋，与问题非相关的信息会稀释相关信息的作用，导致相关信息的有效性减弱。

专栏 2.8　怎样公布好消息和坏消息

上市公司如何发布消息甚至可以影响股市的走势，如果说一家公司要向它的股东公布一项好消息，那么以什么样的方式公布这个好消息才能使它产生最积极的效果呢？如果要公布的是一项坏消息，公司又该如何做才能最大限度地减少其不利影响呢？芝加哥大学行为科学教授塞勒提出了以下四个原则。

（1）如果你有几个好的消息要发布，应该把它们分开发布。例如假定今天你老板奖励了你 1 000 元钱，而且你今天在一家百货商店抽奖的时候还抽中了 1 000 元钱，那么你应该把这两个好消息分两天告诉你妻子，这样的话她会开心两次。根据前景理论，分别经历两次获得所带来的高兴程度之和要大于把两个获得加起来经历一次所带来的总的高兴程度。还如，在你给人送两件以上生日礼物时，不要把所有礼物放在一个盒子里，应该分开包装；若你是老板，给人一次性发 5 000 元，不如先发 3 000 元，再发 2 000 元。

（2）如果你有几个坏消息要公布，应该把它们一起发布。比方说如果你今天钱包里的 1 000 元钱丢了，还不小心把你妻子的 1 000 元钱的手机弄坏了，那么你应该把这两个坏消

息一起告诉她。因为根据前景理论,两个损失结合起来所带来的痛苦要小于分别经历这两次损失所带来的痛苦之和。比如开会收取会务费时,务必一次收齐并留有余地,若有额外开支一次次增收,虽然数量不多,会员仍会牢骚满腹。

(3) 如果你有一个大大的好消息和一个小小的坏消息,应该把这两个消息一起告诉别人。这样的话,坏消息带来的痛苦会被好消息带来的快乐冲淡,负面效应也就少得多。

(4) 如果你有一个大大的坏消息和一个小小的好消息,应该分别公布这两个消息。这样的话,好消息带来的快乐不至于被坏消息带来的痛苦所湮没,人们还是可以享受好消息带来的快乐。具体情况具体分析:如果差距过大应分开;差距不大应该整合。

本章小结

(1) 认知过程是个体认知活动的信息加工过程。

(2)“有限理性”的概念,个体在做决策时远远达不到经济学理论中假定的“最大化”和“最优”,而只能达到“满意”。

(3) 启发式是凭借经验解决问题的方法,是一种思考捷径,是解决问题的简单而笼统的规律或策略,也被称为经验法则或拇指法则。

(4) 代表性启发式是指人们倾向于根据样本是否代表(或类似)总体来判断其出现的概率,代表性越高的样本对其判断的概率也越高。

(5) 可得性启发式,是指人们倾向于根据客体或事件在知觉或记忆中的可得性程度来评估其出现的相对频率,容易知觉到的或回想起的客体或事件被判定为更常出现。

(6) 锚定和调整启发式,是指在判断过程中,人们对于不确定数值的估计往往是基于对初始值或者起始点进行适当调整的结果。

(7) 框定是人们解释外在客观世界的心理模式,用来作为了解、指认以及界定行事经验的基础。

由于描述方式的改变而导致选择偏好发生改变的现象称为框定效应。

框定依赖偏差,是指人们的判断与决策依赖于所面临的决策问题的形式,即尽管问题的本质相同,但因形式的不同也会导致人们作出不同的决策。

思考题

1. 请比较理性、有限理性和生态理性三个概念的区别。
2. 简述代表性启发式的含义及其造成的偏差。
3. 简述可得性启发式的含义及其造成的偏差。
4. 简述锚定和调整启发式的含义及其造成的偏差。
5. 人们了解了各种启发式偏差以后,在生活中能否避免犯同样的错误?

第3章

投资者心理与行为偏差

引导案例：长生生物退市前连续涨停

2018年7月5日，根据举报提供的线索，国家药监局对长春长生生物科技有限公司进行飞行检查。7月15日，国家药监局发布《关于长春长生生物科技有限责任公司违法违规生产冻干人用狂犬病疫苗的通告》，指出长春长生冻干人用狂犬病疫苗生产存在记录造假等行为。

长春长生是深交所上市公司长生生物(002680)主要的子公司。7月23日，因涉嫌信息披露违法违规，长生生物被中国证监会立案调查。该公司当天发布公告表示，如果公司被监管部门最终认定存在重大违法行为或移送公安机关，根据有关规定，公司股票可能存在被实施退市风险警示、暂停上市或终止上市风险。同一天，深交所对长生生物大股东、董监高所持有的股份进行限售处理。7月26日，受疫苗事件影响，长生生物主要银行账户被冻结，开市起被实施其他风险警示，股票简称变更为"ST长生"。

10月16日，长春长生因违法违规生产疫苗，被国家药品监督管理部门作出吊销药品生产许可证的行政处罚决定，并处罚没款91亿元，上述罚没金额为长生生物2015年—2017年营业收入合计33.67亿元的2.7倍。11月5日长生生物被实施退市风险警示，其简称再度变更为"*ST长生"。

自7月16日至8月29日，长生生物连续32个跌停，期间市值蒸发207.27亿元。但从11月8日至16日，公司股价连续7个涨停。Wind数据显示，涨停期间，机构、大户、中户、散户资金流入额合计17.06亿元，其中中户、散户资金大约占比70%。

2018年11月16日深夜，沪深交易所发布《上市公司重大违法强制退市实施办法》，同时启动对长生生物重大违法强制退市机制。随着退市新规落地，七连板疯涨后的长生生物，被深交所启动重大违法强制退市机制，17亿元入场资金难以出逃。

在长生生物疫苗事件过程中，相关部门和长生生物公司一再向投资者警示长生生物

退市的风险,但是仍然有部分投机者无视这种风险,在长生生物面临强制退市前,还在对其股票进行疯狂投机。这些投机者的行为,很难得到市场的理解。在这些投机者中间,有些人带着投机和赌博的心理,认为长生生物有可能不会被真正强制退市;有些人则是跟风操作。实际上,对于长生生物而言,因其事件造成的社会影响很大,社会舆论对其关注度非常高,因此最后不被强制退市的可能性微乎其微;而且被强制退市后几乎不可能再次回归A股,重新上市的概率几乎为零。显然,这些投机者的动机和行为完全是非理性的。

2019年1月14日,深交所发布《关于对长生生物科技股份有限公司股票实施重大违法强制退市的决定》(深证上〔2019〕23号),决定对长生生物实施重大违法强制退市,长生生物将从2019年3月15日起开始暂停上市。

投资者在进行投资决策时,其心理因素会随着外界环境的变化而发生微妙的改变,人类固有的行为模式会不知不觉地主宰着投资者的行为,尤其是当证券市场面临着不确定性和不可预测性时,情况更是如此。每个投资者开始总是试图进行理性决策,并希望规避风险,但是,当投资者发现由于自己的有限能力不能把握投资行为的可靠性时,投资者就会向政策的制定者、媒体、专家或者自己的感觉、经验等寻求心理依托,投资行为的前景越不明朗,投资者的心理依托感就会越强,人类特有的认知偏差的弱点就会更为明显。

本章从人类心理角度,分析投资者在面临不确定性条件下的决策过程中出现的心理和行为偏差。其主要表现为过度自信、损失厌恶(loss aversion)所导致的禀赋效应和短视偏差,同时,在不确定性决策过程中,投资者还会出现证实偏差、后悔厌恶、模糊厌恶、心理账户(mental accounting)、时间折扣等偏差。这些偏差的存在,影响着人类的理性决策。由于行为金融理论和实证对过度自信的研究结论相当丰富,在本书第7章将专门介绍过度自信及其对金融决策产生的影响。

3.1 心理账户

1. 心理账户的含义与特点

心理账户是芝加哥大学行为科学教授理查德·塞勒提出的概念。他认为,除了钱包这种实际账户外,在人的头脑里还存在另一种心理账户。人们会把在现实中客观等价的支出或收益在心理上划分到不同的账户中,根据资金的来源、资金的所在和资金的用途等因素对资金进行归类,这种现象被称为“心理账户”。塞勒认为:小到个体、家庭,大到企业集团,都有或明确或潜在的心理账户系统。在做经济决策时,这种心理账户系统常常遵循一种与经济学的运算规律相矛盾的潜在心理运算规则,其心理记账方式与经济学和数学的运算方式都不相同,因此经常以非预期的方式影响着决策,使个体的决策违背最简单的理性经济法则。

心理账户的特点是:①每一项资金(资产或事情)都被放入一个心理账户。②各个心理账户通常是独立的、被分割的。③人们通过心理预算来考察每一个心理账户的成本与收益是否匹配,或者做一件事情是否划算。在进行决策时,个体通常并不从全局进行考虑,而是在心里无意识地把一项决策分成几个“心理账户”,对于每个心理账户人们会有不

同的决策。

心理账户说明人在金钱面前是很主观的、非理性的，并不具有经典经济学或金融学所宣称的理性。例如，传统经济学认为同样是100元，是工资挣来的，还是彩票赢来的，或者路上捡来的，对于消费来说，应该是一样的，可是事实却不然。在人们心里，会把辛苦赚来的钱和意外获得的钱放入不同的账户。人们会对辛苦赚来的钱有严谨的储蓄和投资计划，但是对意外获得的钱却持有相对随意的态度。对于理性的人而言，其名下的钱并不因为它的来源而有性质上的区别。

我们可以通过卡尼曼和特沃斯基(1981)设计的实验来进一步了解心理账户。

情形1：你打算去剧院看一场演出，票价是10美元，在你到达剧院的时候，发现自己丢了一张10美元钞票，你是否会买票看演出？实验结果表明(调查对象为183人)：88%的调查对象选择“会”，12%的调查对象选择“不会”。

情形2：你打算去看一场演出而且花10美元钱买了一张票，在你到达剧院的时候，发现门票丢了。如果你想看演出，必须再花10美元，你是否会买票？实验结果表明(调查对象为200人)：46%的调查对象选择“会”，54%的调查对象选择“不会”。

卡尼曼和特沃斯基认为：两种实验情境出现明显不同结果的原因在于：在考虑情形1的决策结果时，人们把丢失的10美元和买演出票的10美元分别考虑；而在情形2，则把已经购买演出票的钱和后来买票的钱放在同一个账户估价，看上去好像要花20美元看一场演出，一部分人觉得“太贵了”，因而改变自己的选择。

2. 心理账户的应用

1) 价格感知

特沃斯基和卡尼曼(1982)通过设计以下情境实验引入“心理账户”与消费者购买决策行为的研究。

情形1：假定你要买一件夹克和一个计算器。在某商场夹克的价格是125美元，计算器的价格是15美元。这时候有人告诉你，开车20分钟后另一个街区的一家商场计算器的价格是10美元。请问：你会去另一个商场买计算器吗？

情形2：假定你要买一件夹克和一个计算器。在某商场夹克的价格是15美元，计算器的价格是125美元。这时候有人告诉你，开车20分钟后另一个街区的一家商场计算器的价格是120美元。请问：你会去另一个商场买计算器吗？

在这两个情境中，其实都是对“是否开车20分钟从140美元的总购物款中节省5美元”作出选择。然而，实验对象在两个情境中的回答却不一样，在情形1中，68%的实验对象选择去另一家商场；而在情形2中，只有29%的实验对象选择开车去另一家商场。选择偏好发生了逆转。

特沃斯基和卡尼曼提出，消费者在感知价格的时候，是从三个不同的心理账户进行得失评价的：一个是最小账户(minimal account)，就是不同方案所享有的优惠的绝对值，在本实验中的最小账户就是5美元。另一个是局部账户(topical account)，也可称为相对值账户。例如，在实验情形1中开车前往另一家店的“局部账户”表现为计算器价格从15美元降为10美元(相对差额为1/3)；而在实验情形2中的“局部账户”表现为计算器价格从

125 美元降为 120 美元(相对差额为 1/25)。第三个是综合账户(comprehensive account),综合账户就是总消费账户,该实验的综合账户为 140 美元。

特沃斯基和卡尼曼认为,在上面的实验中,消费者是自发运用了局部账户,即通过相对优惠值来感知价格,情形 1 有 33.3%的优惠,而情形 2 仅有 4%的优惠,因此,人们的购买行为发生了反转——表现为在实验情形 1 中,68%的实验对象选择去另一家商场;而在实验情形 2 中,却只有 29%。

此后,Moon,Keasey 和 Duxbury 对特沃斯基和卡尼曼的研究进行了重复实验并且提出,当优惠超过某个阈限值的时候,消费者对绝对优惠值同样非常敏感。消费者对绝对值和相对值优惠的体验呈现这样一种规律:当购买额度较小时,消费者对相对值优惠更敏感,即在商家让利水平相同的情况下,相对值能让消费者体验到更多的优惠,所以,对额度小的商品,商家应呈现相对值优惠。随着购买金额的增加,绝对值与相对值之间的优惠体验差距会逐渐缩小,如果继续增加购买金额,优惠体验就会出现相反的结果,此时消费者对绝对值更敏感,即商家通过呈现绝对值优惠更能促使消费者购买。

2) 心理账户在金融投资决策领域的应用

心理账户在金融投资决策领域最广泛的应用是投资组合结构的运用。根据理性投资组合理论,投资者应该只关心他们投资组合的期望收益,而不应该关注某个特定投资部分的收益。可事实相反,投资者倾向于把他们的资金分成安全账户(保障他们的财富水平)和风险账户(试图做风险投机的买卖)。1997 年 Fisher 和 Statman 提出:人们在投资时会把资金分别放在不同的投资账户中,即使是基金公司也建议投资者建立一个资产投资的金字塔,把现金放在金字塔的最低层,把基金放在中间层,把股票放在金字塔的最高层。

2000 年,Shefrin 和 Statman 提出了多重账户行为资产组合理论(behavioral portfolio theory multiple mental account,BPT-MA)。多重账户行为资产组合理论是建立在卡尼曼和特沃斯基的前景理论之上的一个框定体系,它认为投资者的资产结构应该是金字塔式的分层结构(这里的"层"就是心理账户),投资者对其资产进行分层管理,每一层对应投资者的一个目标。底层是投资者为避免贫穷而设立的,所以,其投资对象通常是短期国债、大额可转让存单、货币市场基金等有稳定收益、风险小的证券;高层是为使其富有而设立的,其投资对象通常是外国股票、成长性股票、彩票等高风险、高收益证券。

专栏 3.1 "支付分离"与会费策略

预先的支付可使人们对购买与消费的感受分开,在消费时则减少了痛苦的体验。较典型的例子是人们的电话费包月偏好,当月初交纳了月租,人们在当月打电话时则没有明显的在开销的感觉,也因此实际消费常常会超出话费预算。另外,健身俱乐部的年费并非按次收取,而是年初一次性收取,它将使用与支付分开,人们在使用时并没有明确地感觉到在消费,而且很多消费者认为他们会经常使用这个收费,是很合算的,如果每次去健身都收费则不能与消费者的这种心理感受相匹配。因此,服务性的商家,如健身、美容,如果采用会费制度,同时把会费制度每次的花费与按次收费的费用进行对比,会吸引更多的消费者,而且因为接受服务时没有明显的花费感受,这可以带给消费者更愉快的消费体验。

3.2 损失厌恶与禀赋效应

1. 损失厌恶的概念

期望理论的重要发现之一是人们在面对收益和损失的决策时表现出不对称性。卡尼曼和特沃斯基(1979)通过实验得出结论:人们并非厌恶风险,当人们认为合适时,他们会选择赌上一把,但如果不是厌恶风险又是什么呢?他们认为,人们的动机主要是躲避损失,而不是那么厌恶风险,人们厌恶的是损失,损失与收益相比,总显得更突出、让人感受更强烈。损失厌恶是指人们面对同样数量的收益和损失时,损失令他们产生更大的情绪影响,实验结果发现同量的损失带来的负效用为同量收益的正效用的2.5倍。期望理论认为,损失厌恶反映了人们的风险偏好并不是一致的:当涉及的是收益时,人们表现为风险厌恶;当涉及的是损失时,人们则表现为风险寻求。

关于损失厌恶的例子是Camerer等(1997)对纽约出租车司机的研究。这些出租车司机需要交一个固定的费用才能获得出租车12个小时的运营权,作为报酬他们可以将这期间的所有所得当作自己的收入。司机必须决定每天工作多长时间。一个最大化的策略是:在较好的日子里(每小时有更多收入的日子,就像下雨天或者在本市有大型会议的日子)工作久一些;在不好的日子里早点收班。然而,假设司机为每天设定了一个收益水平作为目标,把不到这个目标的收入当作损失,那么他们应该在较好的日子里较早收班,在不好的日子里较晚收班——恰恰与理性的策略相反。这就是Camerer等在实证研究里的发现。

2. 禀赋效应

捕猎野鸭是美国人最热衷的一项狩猎活动,但是随着人类对自然环境的影响不断加剧,野鸭的数量不断减少。为了保护野鸭的生息和繁衍,要么对捕猎者的人数进行限制,要么让捕猎者承担对野鸭栖息湿地的保护成本。美国学者哈马依克和布朗(Hannaek和Brown,1974)就此做了一个调查,结果发现,捕猎者愿意平均每人支付247美元的费用以保护适合野鸭生存的湿地环境,但是如果让他们放弃在湿地上捕猎野鸭,他们要求的赔偿却高达平均每人1 044美元。也就是说,让这些捕猎者放弃捕鸭的权利比让他们购买捕猎野鸭的资格要难得多。这种个体在拥有某物品时对该物品的估价高于没有拥有该物品时的估价的现象被称为“禀赋效应”(endowment effect)。

塞勒(1980)首次提出了禀赋效应的概念,认为禀赋效应导致了WTA(willingness to accept,愿意接受的价格)和WTP(willingness to pay,愿意支付的价格)之间的差异,并将其定义为:与得到某物品愿意多支付的金额相比,个体出让该物品所要求得到的金额通常更高。后续的深入研究表明,禀赋效应是一种相对稳定的个体偏好,具有跨物品、跨时间、跨年龄和跨情境的稳定性,是一种极为普遍的心理现象。

禀赋效应是与损失厌恶相关联的现象。许多决策是在两种方案中选择:维持现状,或者接受一个新的方案(其在某些方面有利而在另一方面不利)。可以将现状视为参考水平,因为损失比收益显得更让人难以忍受,所以决策者偏爱维持现状。由于禀赋效应,人

们不愿意放弃现状下的资产，投资者偏好于持有他们已有的东西而不愿意与别人进行交易以换得更好的替代品。因为放弃一项资产的痛苦程度大于得到一项资产的喜悦程度，所以个体行为者为了得到资产的“支付意愿”要小于因为放弃资产的“接受意愿”。例如，许多人不愿意用自己的彩票去和别人同样的彩票外加部分现金进行交换。

禀赋效应导致买卖双方心理价格出现偏差，从而影响市场效率。如果让人们对某种经济利益进行定价，则其得到的这种经济利益所愿支付的最大值，远远小于其放弃这种经济利益所愿接受的最小补偿值。或者说，同样一个东西，如果是本来就拥有的，那么卖价会较高；如果本来就没有，那愿意支付的价钱会较低。例如，在劳动力市场中，应聘者的薪酬目标与雇主愿意支付的金额存在一定的差距；在城市化进程中，政府部门和居民在拆迁补偿金额方面总是很难达成一致。

禀赋效应也导致了交易惰性(reluctance to trade)。例如，一位受试者假设被指派了一些工作，其在工资(S)和工作环境(T)两方面有差异。该受试者被指派到一个职位(S1，T1)，然后可以选择是否跳槽到另一个职位(S2，T2)。(S2，T2)相对(S1，T1)而言，在一个方面更好，而另一个方面更差。实验结果表明，原来被指派到(S1，T1)的人不愿意跳槽到职位(S2，T2)，原来被指派到职位(S2，T2)的人也不愿意跳槽到职位(S1，T1)。

虽然禀赋效应的存在会降低市场交易效率，但是它也有很多积极的应用，特别是在商业领域。精明的商家利用禀赋效应引导消费者的购买行为。最典型的例子是对商品的各种“试用”，其次是网络购物中购物车的设置，都可以令消费者产生已经“拥有”商品(虽然很短视)的错觉。

3. 短视损失厌恶

1995 年，Benartzi 和塞勒进一步提出短视损失厌恶(myopic loss aversion，MLA)的现象，认为该现象可以用于解释股票溢价之谜，并于 1996 年证实该现象的存在。他们认为，在股票投资中，长期收益可能会周期性地被短期损失打断，短视的投资者把股票市场视同赌场，过分强调潜在的短期损失，投资者不愿意接受这种短期损失的现象被称作“短视损失厌恶”。

Benartzi 和塞勒认为如果损失厌恶的投资者不去频繁评估他们的投资绩效，他们会更愿意承担风险，即评估期越长，风险资产越有吸引力，风险溢价也就越低，然而，现实的客观和主观因素导致投资者的评估期总是有限的。例如，从客观因素来讲，资产收益的概率分布并不是长期稳定不变的，有时甚至还会因破产、财务危机等巨变而发生剧烈变化，投资者需要根据所获取的信息重新评估资产收益概率分布的变化，以调整投资决策，这样，资产收益的概率分布的稳定性以及可获取信息的频率(如年报等定期报告)就会影响投资者的评估期；此外，投资者的消费计划、风险偏好等主观因素也会影响其自身的评估期选择，在这种情况下，MLA 理论能够对投资者的行为予以合理的解释。

1999 年 2 月，以色列最大的共同基金管理人 Hapoalim 银行宣布改变其披露政策，将每月披露一次业绩改为每 3 个月披露一次，这样就使得那些依赖从银行获得业绩信息的投资者获取信息的频率下降，从而使投资者更愿意持有其基金。银行的理由是“投资者不应该为偶然的价格下跌所惊扰”，这也成为 MLA 理论的一次实践应用。

MLA 当中的“短视”就是指人们在短期中评估资产组合的行为。短视对投资行为的影响可以用重复博彩和单次博彩的差别来解释，Samuelson 首先关注了这一问题。他先设计了一个预期收益大于零的赌博——50％ 的机会赢得 200 美元，50％ 的机会损失 100 美元。然后问一位同事是否愿意参加这个赌博。这位同事拒绝了这个赌博，因为他感觉可能的损失导致的效用降低会大于可能的收益带来的效用增加，但是他乐于接受 100 次这样的赌博。这个例子一方面隐含着损失厌恶的概念；而另一方面说明，大数定律会增加赌博的确定性，提高人们承担风险的能力。这说明，重复博彩会增大风险资产的吸引力；同时也说明，重复博彩问题并不是投资者非理性的表现，而是投资者对风险资产收益的概率分布的理性反应。

Benartzi 和塞勒提出的 MLA 理论引起了理论界的极大兴趣，许多学者对于这个理论本身是否正确进行了研究，绝大多数的研究是通过心理实验来直接验证人们这种心理特征是否存在。这些研究不但证实 MLA 在不同的实验环境及团队决策下都普遍存在，而且增进了人们对于信息在 MLA 中的作用的理解，提高了对 MLA 的理论认识水平。

塞勒、特沃斯基和 Schwartz 运用心理实验的方法为此提供了直接证据。他们设计了如下的实验。实验对象可以投资于两个基金：一个是与实际的某 5 年期债券相对应的低风险基金，另一个是模拟股票指数基金的高风险基金，两个基金的预期收益都为正；实验被试必须通过实际体验了解风险和收益的分布。通过对 1 个月、1 年以及 5 年评估期的比较发现，评估期的加长可以减少损失的体验，因而提高对风险资产的投资比例。

Gneezy 和 Potters 也设计了一个相似的实验，得出了相同的结果。实验分为 12 轮，参加者在每个反馈期期初获得 200 美分的初始禀赋并进行投资；风险资产为有 2/3 的可能性损失全部投资、1/3 的可能性赢回 3.5 倍投资数额的彩票，参加者完全了解资产的概率分布。参加者可以考虑两种处理方式：一种是“高频率”处理方式，实验对象可以决定在每一轮投资多少，并在每一轮之后得到收益的反馈；另一种是“低频率”处理方式，实验对象可以每 3 轮决定他们的投资数额并得到收益反馈。在“低频率”处理方式中，实验对象对风险资产的投资明显比“高频率”处理方式多。这说明，更长的投资评估期使有正预期收益的风险资产的吸引力增大。

在早期的这两个实验当中，“信息反馈频率”和“投资调整周期”被用作“短视”的代理变量，前者体现了信息对短视的影响，而后者则体现了投资调整行为对短视的影响。

3.3 证实偏差

人们一旦形成一个信念较强的假设或设想，有时会把一些附加证据错误地解释为对他们有利，因而不再关注那些否定该设想的新信息。人们有一种寻找支持某个假设的证据的倾向，这种证实而不是证伪的倾向称为“证实偏差”(confirmation bias)。

证实偏差产生的原因有以下几个：①信念坚持(belief perseverance)是导致证实偏差的心理基础，他们会坚持相信他们的假设，即使这个假设和新数据相矛盾。这种偏差导致对新数据没有足够重视。②锚定往往也是导致证实偏差的心理因素之一。锚定导致人们

忽视附加证据。心理学研究揭示了这样的现象,人们倾向于把肯定的证据视为相关而且可靠的,而将否定的证据视为不相关而且不可靠的,因此在价值判断中,容易接受肯定的证据,忽略否定的证据。③对资料的选择性收集或审查。

证实偏差是金融市场正反馈机制形成的推动力之一。在金融市场上,当整个市场处于繁荣的上升期,即使有各种各样的证据表明市场已经被严重高估,人们仍然会倾向于忽略这些负面的信息,而对正面信息赋予更高的权重,乐观地推动市场价格越来越偏离其基础价值;相反,当市场处于低迷时期,人们会倾向于对负面信息赋予更高的权重,从而导致市场更加低迷。

企业经理人的证实偏差则更多地表现在投资决策上,在所投资的项目产生问题时,对负面信息不予重视,仅关注正面信息;在应当终止某个失败项目的时候,依然继续向项目追加资金,最终导致决策错误。

专栏3.2　工业大麻概念股

2019年1月27日晚间,康恩贝发布公告称,公司全资子公司希陶公司的全资子公司云杏公司收到加工大麻花叶项目申请批复。同时,康恩贝表示,随着生物技术发展,工业大麻成为全球新的产业热点,云杏公司申请开展大麻花叶加工项目是公司对工业大麻领域的全新探索,在该项目建成并取得加工许可证后,公司将对工业大麻中的大麻二酚(CBD)及其他活性成分的提取物做加工技术研究和产品研发,为公司寻求新的利润增长点。

和康恩贝相比,顺灏股份掀起的"波澜"更大。在其于1月17日公布获得种植许可证以及加工大麻花叶项目申请的批复后,股价已经从4元多上涨到近10元。据统计,A股12只工业大麻概念股自1月到3月以来,没有一只股票涨幅低于30%,10家涨幅超50%,4家涨幅超过200%。金鹰股份主要业务为麻、毛、丝、绢纺织机械成套设备制造销售,但也被当作大麻概念股来炒,3月20—22日连续三个涨停。无奈之下,公司3月23日发布风险提示公告,与工业大麻概念并无关联。

工业大麻概念股热炒引起相关部门关注,国家禁毒委员会办公室于3月27日下发了《关于加强工业大麻管控工作的通知》。通知提出,《1961年麻醉品单一公约》规定,工业用大麻限于纤维和种子,其他用途的种植排除在外,并要求各省市自治区禁毒部门要严把工业大麻许可审批关,并声明我国目前从未批准工业大麻用于医用和食品添加。通知给热炒的"工业大麻"概念泼了一盆"冷水"。

康恩贝向投资者发出了四大风险提示,称公司工业大麻加工业务在法规政策、监管要求、自身实施等方面存在不确定性。康恩贝表示即使云杏公司试制和改造如期顺利进行,并且试制完成后经公安部门检查合格,何时和能否最终取得公安机关核准颁发的正式加工许可证也存在不确定性。

即使监管部门和上市公司都向投资者发出了风险提示,不少投资者依然相信工业大麻概念会给公司利润带来新的增长点,体现出所谓的"证实偏差"。在康恩贝上证e互动平台上,直至2019年4月,绝大部分投资者的问题依然与工业大麻有关,即使这些问题在上市公司公告里已经澄清。

3.4 后悔厌恶与处置效应

1. 后悔厌恶

后悔是一种非常普遍而且非常容易理解的心理，人们常常为作出了错误的决策而自责不已，这种情绪就是后悔。显然，后悔是一种消极的情绪。损失会让人很痛苦，而后悔是一种除了损失之外，还自认为必须对此负责任的感受，因此后悔比损失更让人痛苦。自豪是后悔的另一面。心理学家和经济学家认识到后悔与自豪对金融决策的重要影响，研究者认为人们会极力避免出现后悔这种感觉，重要的是，自豪和后悔的影响并不对称，人们对后悔有着强烈的抵触。

后悔厌恶就是指人们为了避免决策失误带来的后悔的痛苦，常常会做出一些非理性的行为。例如，投资者趋向于获得一定的信息后，才作出决策，即便是这些信息对决策来讲并不重要，没有它们也能作出决策，这主要是为了减少后悔给自身带来的精神损失。

后悔厌恶理论的核心是以下三个定理。

定理1：胁迫情形下采取行动所引起的后悔比非胁迫情形下的后悔要轻微。

定理2：没有做错误的行为引起的后悔比做了错误的行为引起的后悔要轻微。

定理3：个体需对行动的最终结果承担责任情形下引起的后悔比无须承担责任情形下的后悔要强烈。有利的结果会使责任者感到自豪，不利的结果会使责任者感到后悔。

前景理论发现，人们有时会寻求冒险，比如面对损失时，或许投资者正是因为想避免失败带来的后悔情绪，所以偏离了他们的风险规避本能，而变得风险偏好。另外，基于后悔厌恶的心理因素，投资者在买卖股票时，会表现出持亏卖盈的行为偏好，研究者发现，投资者会过早地卖掉表现良好的股票而又长期地持有亏损的股票。

2. 处置效应

处置效应(disposition effect)，是指投资人在处置股票时，倾向于卖出赚钱的股票、继续持有赔钱的股票，也就是所谓的“出赢保亏”效应。处置效应的表现为，当投资者盈利时，面对目前确定的收益和未来不确定的走势时，为了避免价格下跌而带来的后悔，倾向于风险回避而做出获利了结的行为。当投资者出现亏损时，面对目前确定的损失和未来不确定的走势，为避免立即兑现亏损而带来的后悔，倾向于风险寻求而继续持有股票。正是由于这种心理，导致人们长时间地持有亏损的股票。

前景理论中的参考点能够对处置效应进行有效的解释。以股票买价为参考点，在收益区域，股票未来价格同等金额的涨或跌所引起的价值变化是不相同的，收益下降所带来的价值下降大于收益上升带来的价值上升，为了避免价格下降带来的恐惧，人们倾向于卖掉股票；相反，在损失区域，收益上升所带来的价值上升大于收益下降带来的价值下降，于是人们倾向于持有股票等待价格的上涨。

3.5 模糊厌恶与本土偏差

1. 模糊厌恶

模糊厌恶是指人们厌恶主观的或模糊的不确定性的程度超过对客观不确定性的厌恶。当这种模糊性可能导致未来巨大损失时，人们对模糊性的厌恶和回避心理就更加强烈了。模糊厌恶在股票投资中表现为：本国偏好、本土偏好。杨晓兰等(2016)对东方财富网论坛股吧的研究显示，投资者更喜欢讨论本地的股票。

以下是一个经典的模糊厌恶的实验。

假定有两个瓶子1号和2号，2号瓶共装有100个球，其中50个红球、50个蓝球。1号瓶也装有100个球，其中有红球也有蓝球，但是参加实验者不知道红球与蓝球的比例，要求参加实验者在以下两个风险事件中进行选择。

A1：从1号瓶抽取一个球，红球得100美元，蓝球得0美元。

A2：从2号瓶抽取一个球，红球得100美元，蓝球得0美元。

实验结果，人们普遍对A2的偏好多于A1。

2. 本土偏差

许多研究表明，投资者尤其是个人投资者在分散化投资时有“本土偏差”(home bias)的倾向，即投资者将他们的大部分资金投资于本国，甚至本地的股票，而不是如马克威茨的投资组合理论所认为的那样，投资于与本地股相关度低、能够降低系统性风险的外地股甚至是外国股票，从而降低投资组合的风险。

本土偏差的表现：一是投资于国内股票。投资者在投资中将绝大部分资金投资在国内市场，而只将很小一部分资金投资在海外市场。二是投资于距离近的公司。本土偏差的另一个表现是在国内股市中将大量的资金投资于自己任职的公司或者公司总部距离自己的住宅比较近的公司。《商业周刊》(1997)曾报道，在一些公司中，即使在雇员可以自由选择投资对象时，他们仍然倾向于投资他们本人工作的公司。塞勒和Benartzi(2005)发现：①美国企业的雇员通常倾向于为本公司股票建立单独的心理账户，与其他股票不发生联系；②美国企业的雇员平均将42%的资产投资在本公司股票上(对本企业过度自信)，导致风险过于集中；③由于本身又受雇于本企业，一旦公司破产倒闭，公司雇员通常会损失惨重，面临薪金和股票投资的双重损失。

3.6 跨期选择与时间折扣

在伊索寓言中，蚂蚁和蚱蜢是一对朋友，但是二者过着完全不同的生活，蚱蜢每日唱歌跳舞，享受温暖的阳光，不思来日；相反，蚂蚁每天忙着为即将到来的冬天储存食物。面对今天和未来，蚱蜢选择享受今天，蚂蚁选择为未来生存而努力工作，这就是典型的跨期选择问题。

跨期选择是指人们对发生在不同时间点的收益或损失进行权衡，从而作出选择与决策。例如，你愿意今天获得 800 元，还是一年后获得 1 000 元？跨期选择不仅仅发生在收益情境中，也发生在损失情境中，行为学研究中已发现了跨期选择的符号效应，即人们在对未来损失的折扣率低于对未来获得的折扣率。

跨期选择领域的研究发现，与当前或近期的奖赏或损失相比，人们常常倾向于低估未来奖赏或损失的价值，这种现象就是所谓时间折扣(time discounting)，又称时间贴现。

为了进一步量化时间折扣在心理意义上的变化，学者们使用时间折扣(贴现)率来衡量时间折扣程度的大小。该领域的焦点问题是，是否存在一个恒定的折扣率或折扣函数。1937 年，经济学家 Samuelson 基于期望效用理论在论文《关于效用的测量》中提出了一个指数贴现效用(Discounted—Utility，DU)模型，用以解释时间折扣问题。DU 模型假设，人们对未来不同时间点的效用按照同一比率(指数函数的形式 r)进行折扣，而折扣后的未来各时期效用之和，就是人们对该决策赋予的总效用。所以，在 DU 模型中时间折扣率是一个恒定值。长期以来，该模型被视为解释时间折扣现象的基本模型。

但是自 20 世纪 80 年代以来，随着行为经济学的兴起，研究者逐渐发现了违背 DU 模型的多种异象。例如，人们对收益的时间折扣率大于损失的时间折扣率(符号效应，sign efect)；奖赏数量不同，折扣率也会产生变化等等。因此，一些研究者在 DU 模型的基础上提出了各种双曲线折扣模型(hyperbolic discounting model)，来解释时间折扣中发现的种种异象。双曲线折扣模型的特征是，人们在短期和长期的折扣(贴现)比率是不同的。

时间折扣的一种表现是时间偏好。时间偏好(time preferences)是指人们倾向于推迟执行那些需要立即投入而报酬滞后的任务，而马上执行那些能立即带来报酬而投入滞后的事情，即相对于远期的奖赏或报酬，人们更偏好即时的奖赏或报酬。在塞勒(1981)的实验中，参与者获得 15 美元的奖金，他们可以选择立即拿到 15 美元，也可以选择等待一段时间获得多于 15 美元的钱。塞勒要求参与者写出他们愿意为推迟获得收入而要求补偿的金额，延长的时间段分别为 1 个月、1 年、10 年。实验结果显示，参与者对应要求补偿的平均金额分别为 20 美元、50 美元、100 美元，按复利贴现计算的贴现率分别为 354%、126%和 19%，如表 3.1 所示。

表 3.1 塞勒的实验结果

	1 个月	1 年	10 年
补偿金额/美元	20	50	100
贴现率/%	354	126	19

近年来，时间折扣问题逐渐成为经济学家、行为决策学家、神经科学与心理学家、精神病学家等学科研究者共同关注的一个问题。这一关注包括两个层面的原因。

其一，人类与动物的重要差异之一是人类具有对未来进行计划和规划的能力，但在进行跨期选择时，人类却同寓言中“朝三暮四”的猴子一样，更偏好即刻的价值。大量神经科学和心理学研究已经从进化的角度证实，从动物到人类，从儿童到成人，普遍存在这种偏好。

其二，由于时间折扣的潜在机制与个体的冲动性和自我控制功能密切相关，时间折扣

可能与一系列精神疾病相关联，如各类成瘾行为、孤独症、暴饮暴食症等。已有大量行为学研究发现，成瘾病人的时间折扣率显著高于正常人，是正常人的 2 倍。也有研究发现人们对于工作的拖延，是由于人们对消费娱乐产品的贴现率高于对工作奖励的贴现率。

McClure 等(2004)首次使用功能磁共振成像技术研究跨期选择的神经基础，发现人们对短期与长期偏好之间选择的不一致性，反映了二者背后不同的神经系统的激活。具体来说，短期的决定是由大脑边缘系统所控制的，大脑边缘系统与中脑多巴胺系统相连，它更容易受到即刻奖赏的影响，而对未来奖赏的价值不敏感。但是，长期忍耐则是由前额叶(PFC)和顶叶皮层所控制，它负责对遥远的将来做长期预期，能够对抽象奖赏和长久延迟的奖赏进行评价。McClure 等的研究从神经科学的角度支持了双曲线折扣模型，揭示了人们在权衡即刻奖赏和延迟奖赏时存在不同的神经基础。

对跨期选择和时间折扣的研究不仅与人类心智的起源与机制等基础科学问题密切相关，也与国家公共政策的制定等现实问题相关。

专栏 3.3　美国海湾战争退伍士兵对补偿方案的选择

20 世纪 90 年代的海湾战争之后，美国政府计划减少军队的规模，预算为退伍士兵提供 34 亿美元的退伍补偿。退伍的士兵要求在两种方案中进行选择：第一种是立即得到 2 万美元的一次性退伍补偿；第二种是在今后每年获得一定的补偿金(这些补偿金的贴现值大约是 4 万美元)。超过 90%的士兵选择立即得到一次性的报酬 2 万美元，这些人的贴现率非常高，在 17%～20%。这些士兵的选择给美国政府节约了 17 亿美元。

事实上，如果士兵的确是现在就需要钱，他们可以向银行贷款 2.5 万美元，而且可以用每年获得的补偿金来偿还债务。

本章小结

本章介绍了投资者在决策过程中可能会出现的心理和行为偏差。主要表现为损失厌恶所导致的短视偏差和证实偏差；以及后悔厌恶、禀赋效应、模糊厌恶、心理账户、跨期选择和时间折扣等。这些偏差的存在，影响着人类的理性决策。

思 考 题

1. 名词解释：心理账户、损失厌恶、禀赋效应、证实偏差、后悔厌恶、处置效应、模糊厌恶、本土偏差、跨期选择、时间折扣、时间偏好。

2. 按照心理账户理论的分析，你认为对于高价商品(如汽车)和低价商品(如面包)，在“绝对值优惠”和“相对值优惠”两种促销优惠，应选择哪一种，为什么？

第4章

期望效用理论与前景理论

引导案例：圣彼得堡悖论

尼古拉斯·伯努利（Nicholas Bernoulli）在1713年提出了著名的圣彼得堡悖论（St. Petersburg paradox）。假设有这样一个游戏：连续掷一枚硬币，出现反面游戏继续，出现正面游戏结束。如果第一次出现正面，你将赢得2元；第二次出现正面，你将获得4元；第三次出现正面，你将得到8元……如果第n次才出现正面，你将得到2^n元。那么，你愿意为参与这个游戏而支付的最大代价是多少呢？

由于游戏在第一次就结束的概率为1/2；第二次为1/4；第三次为1/8。在任何次数结束的概率为1=1/2+1/4+1/8+…由此，这个游戏收益的数学期望值是无穷大。期望收益计算过程如下：

$$(1/2)2+(1/4)4+(1/8)8+\cdots=1+1+1+\cdots=\infty$$

而实验的结果却是，人们为得到这一游戏机会愿意支付的金额很少。

1738年，伯努利又做了一个实验，让被试在下述选项中作出选择。

A：有85%的机会得到1 000美元。

B：有100%的机会得到2美元。

他发现绝大多数人更喜欢确定事件而非彩票，虽然彩票有更大的期望收益（彩票的期望收益值等于每个可能结果乘以概率的加权均值，即850美元）。他认为人们不是根据货币结果的期望值，而是根据这些结果的主观价值（效用）的期望值来评价彩票的。因此，伯努利提出了精神价值即效用值的概念，即人们在拥有不同财富的条件下，增加等量财富所感受到的效用值是不一样的，随着财富的增加，其效用值总是在增加，但效用值的增长速度是递减的。

通过圣彼得堡悖论，发现人们在进行决策时，面对巨额的不确定性很大的期望收益，只愿意付出很小的成本，这是无法用期望效用理论来解释的。那么，是什么因素影响了人

们的风险决策？这些因素会给期望效用理论带来哪些挑战？本章将对这些问题进行解答。

期望效用理论建立在个体偏好理性的基础之上，有消除性、传递性、占优性和恒定性等一系列严格的公理化假定，但是期望效用理论与真实行为广泛偏离。随着实验心理学的发展，行为经济学通过实验论证发现的确定性效应、同结果效应、同比率效应和偏好逆转等现象对期望效用理论形成了巨大冲击。许多经济学家从不同侧面提出了对期望效用理论的修正或扩展模型，但是这些修正模型并不十分令人满意。与这些修正模型不同的是，卡尼曼和特沃斯基提出了前景理论，在一定程度上对个体决策与偏好的实验结果提供了合理解释。

本章介绍的主要发现是建立在一系列实验结果之上的。这些实验通常让被试在两个不确定性的"彩票"之间进行选择，以判断被试对这两个彩票的偏好关系。本章的学习目标是，了解期望效用理论的主要内容；了解心理学实验对期望效用理论提出的挑战；掌握前景理论的主要内容；掌握价值函数和决策权重函数的主要特征。在此基础上对不确定条件下人们的选择行为进行更好的解释。

4.1 期望效用理论

1. 期望效用理论的提出

人们在需要作出选择时，往往面对的是不确定性的环境。所谓不确定性，就是指事件的发展有多种可能结果，人们无法肯定会出现哪种结果。许多经济现象与不确定性有着直接和间接的关系，如投资和保险等，现实生活中到处存在不确定性，人们无法预料明天会发生什么偶然性的事件。Campbell，Lo 以及 MacKinlay（1997）曾经指出，金融学最根本的特征是在金融理论和金融实践中，不确定性都具有主导性作用，每个金融模型的起点都是投资者面临的不确定性，每个模型的结果都是不确定性对投资者行为以及市场价格的最终影响。

研究投资者的偏好，以及投资者如何决策和评估风险，是任何试图解释交易行为或者资产价格发生过程的理论和模型都不可缺少的前提条件。决策是人们从多种备选方案或者事件中作出选择；偏好是人们对不同方案或者事件状态进行价值与效用上的辨优。决策和偏好是紧密联系的，它们共同构成了人类一切经济行为的起点，是经济学和金融学研究不可缺少的重要组成部分。

一般而言，在不确定情况下，决策可以分为两种。如果人们仅仅知道可能会出现的结果有哪几种，但却不知道出现的概率是多少，这种情况下的决策被称为模糊型（ambiguity）决策；如果人们不仅知道结果的类型，同时也了解出现各种结果的概率大小，这时决策被称为风险型（risk）决策。对于前者，由于不知道概率，因此不能采用概率的方法计算，而采用诸如好中求好（maximax）、坏中求好（maximin）、等概率（equal probability）、乐观系数（coefficient of optimism）或后悔最小（minimax regret）等准则进行判断；对于后者，往往依据直接后果预期、期望效用或主观期望效用值等，采用统计学的方法进行判断。

期望效用理论建立在个体偏好理性的一系列严格的公理化假定基础之上,是现代决策理论的基石,是关于理性选择最通用的解释,也是传统金融理论和微观经济学理论的核心理论。该理论由冯诺伊曼(Von Neumann)、摩根斯坦(Morgenstern,1947)和萨维奇(Savage,1954)等进行严格的公理化阐述而形成,后来又被阿罗(Arrow)和德布鲁(Debreu)等吸收进瓦尔拉斯均衡的框定中,成为处理不确定性决策问题的分析范式、价值理论的核心及市场均衡的前提。

期望效用函数,即 VNM 效用函数:$EU = \sum U(x_i)p_i$ 。模型的基本内涵是:不确定条件下最终结果的效用水平是通过决策主体对各种可能出现的结果的加权估价后形成的,决策者追求的是加权估价后期望效用的最大化。这一过程与决策者的风险态度密不可分,期望效用函数的无差异曲线斜率越大则风险规避程度越高,斜率越小则风险规避程度越低。

2. 期望效用理论的公理化假设

根据特沃斯基和卡尼曼(1986)的研究,期望效用理论有 4 个重要的假设条件:消除性(cancellation)、传递性(transitivity)、占优性(dominance)和恒定性(invariance),并且这 4 个假设是按照规范要求进行排序的,借助这些假设,可以推导出期望效用理论;除此之外,还有可比性和连续性等技术性要求。

1) 消除性

消除性是指消除那些无论决策者做何选择都会产生同一结果的状态,最后只有一个状态可以实现,从而将不同的选项单独地合理评价。不同选项之间的选择仅仅依赖于其产生不同结果的状态。例如,如果一个方案包含 A 和 B,另一个方案包含 A 和 C,人们在对两个方案进行选择的时候,会首先将 A 从两个选项中删除,仅比较对结果产生不同影响的 B 和 C。消除性有多种正式表达方式,如“替代公理”(substitution axiom,Von Neumann 和 Morgenstern,1947)、“确定事件原则”(the extended sure-thing principle,Savage,1954)、“独立性条件”(the independence condition,Luce 和 Krantz,1971)等。将各种彩票之间的偏好表示为期望效用最大化,消除性是必要的,它与期望效用函数具有的概率线性性质是等价的。

2) 传递性

风险与无风险选择模型的基本假设是偏好的传递性。偏好的传递性假设对于序数效用 u 表示的偏好不仅是必要的,而且基本上也是充分的,只要 $u(A)>u(B)$,则 A 就优于 B;反之亦然;一般地,只要 A 优于 B,B 优于 C,那么 A 就优于 C。如果可以对每个选项赋予一个不依赖于其他选项的值,则传递性假设就能得到满足。因此,当能够单独对每个选项给予评价,传递性可能成立;当一个选项的结果依赖于其他作比较的选项时,传递性就可能不成立。支持传递性的一个常见观点是:循环偏好能支持货币泵(money pump)现象,即一个偏好不满足传递性的人可以通过一系列的付费交易,最终回到其最初的选择。

3) 占优性(优势性)

它是理性选择显而易见的原理,如果一个选项在某种状态下优于另一个选项,而且在所有其他状态下,都至少与另一个选项一样好,则应该选择这个占优的选项。举一个简单

的例子,假设投资者想在两个风险资产 X 和 Y 之间作一个选择,如果在未来任何情况下 X 的收益总是超过 Y 的收益,只要投资者是永远不会满足的,那么投资者不会持有 Y,因为持有 X 得到的回报一定会更好。与前两个假设条件相比,占优性更简单,也更具有吸引力,是期望效用理论的基石。

4) 恒定性(不变性、独立性)

期望效用理论关于选择的一个基本原则是恒定性,它是指同一个选择问题的不同表述应该产生同样的偏好,也就是说,对各种选项的偏好应独立于对它们的描述,不受对选项描述的影响。如果决策者通过仔细考虑,认为某两个特征只不过是同一问题的不同描述而已,这两个特征就应该产生相同的选择,即使这种仔细考虑不会带来额外收益。恒定性原则被人们普遍认同,人们将它默认为公理而不需要验证。这一原则体现了规范理论的直觉知识,即不影响实际结果的形式变化不应影响人们的选择。

期望效用理论的这 4 个假设中,占优性和恒定性是核心,传递性受到质疑,而消除性已经被许多学者所推翻。事实上两个经典的反例:Allais(1953)和 Ellsberg(1961)的研究使一些学者放弃了消除性和一般的期望原则,但许多模型支持传递性、占优性和恒定性,如 Hansson(1975)、Hagen(1979)、Machina(1982)、Quiggin(1982)、Weber(1982)、Chew(1983)、Fishburn(1983)、Schmeidler(1984)、Segal(1984)、Yaari(1984)、Luce 和 Narens(1985)等;也有一些研究放弃了传递性但保留了恒定性和占优性,如 Bell(1982)、Fishburn(1982,1984)、Loomes 和 Sugden(1982)等。这些理论为了保持期望理论作为可描述的模型,而放弃消除性和传递性来弱化标准模型;但是这些策略并不能很好地解释已经证明的对于恒定性和占优性的违背现象,由于恒定性和占优性在标准金融理论中是核心,所以理性选择理论不足以解释人们的决策行为。

3. 风险态度及效用函数

期望效用理论的基本内涵是:风险情境下最终结果的效用水平是通过决策主体对各种可能出现的结果加权估价后获得的,决策者谋求的是加权估价后预期效用的最大化。

存在风险时人们的选择行为取决于决策者对待风险的态度或偏好程度,期望效用理论的一个重要意义是将不确定性和决策主体的风险态度区分开来。概率描述了结果的不确定性,而定义在结果空间上的效用函数的性质描述了决策主体对待风险的态度。可以进一步通过比较期望效用值和收入的数学期望值来说明决策者对待风险的态度。例如,决策者对彩票的估价可以称为彩票的确定性等价,此时彩票的期望效用值和确定性等价的效用值是相等的。风险厌恶者对彩票的估价低于彩票的数学期望值,风险寻求者对彩票的估价高于彩票的数学期望值,而风险中性者对彩票的估价等于彩票的数学期望值。

由于决策主体的选择会影响收入和效用,对风险偏好程度的衡量可以依据决策者对选择而导致的效用变化的态度来衡量。风险的主观态度可以分为三类:风险厌恶(risk aversion)、风险寻求(risk seeking)和风险中性(risk neutral)。

1) 风险厌恶

假定决策者在无风险条件下所能获得的确定性收入与他在有风险条件下所能获得的期望收入相等,如果决策者获得确定性收入的效用高于有风险条件下获得同样期望收入

的效用，或者说他更偏好确定性收入，该决策者为风险厌恶型，或风险规避者。风险厌恶下的效用函数是凹函数，即风险厌恶者的收入—效用关系曲线是凹向原点的。对于风险厌恶者来说，货币收入所提供的总效用是以递减的速率增加，即边际效用递减。

例如，在图4.1中，A 点的期望效用为：$U[pX_1+(1-p)X_2]$，p 为0～1的常数。B 点的期望效用为：$pU(X_1)+(1-p)U(X_2)$，即 $U[pX_1+(1-p)X_2]>pU(X_1)+(1-p)U(X_2)$。

该效用函数满足凹函数的特征。

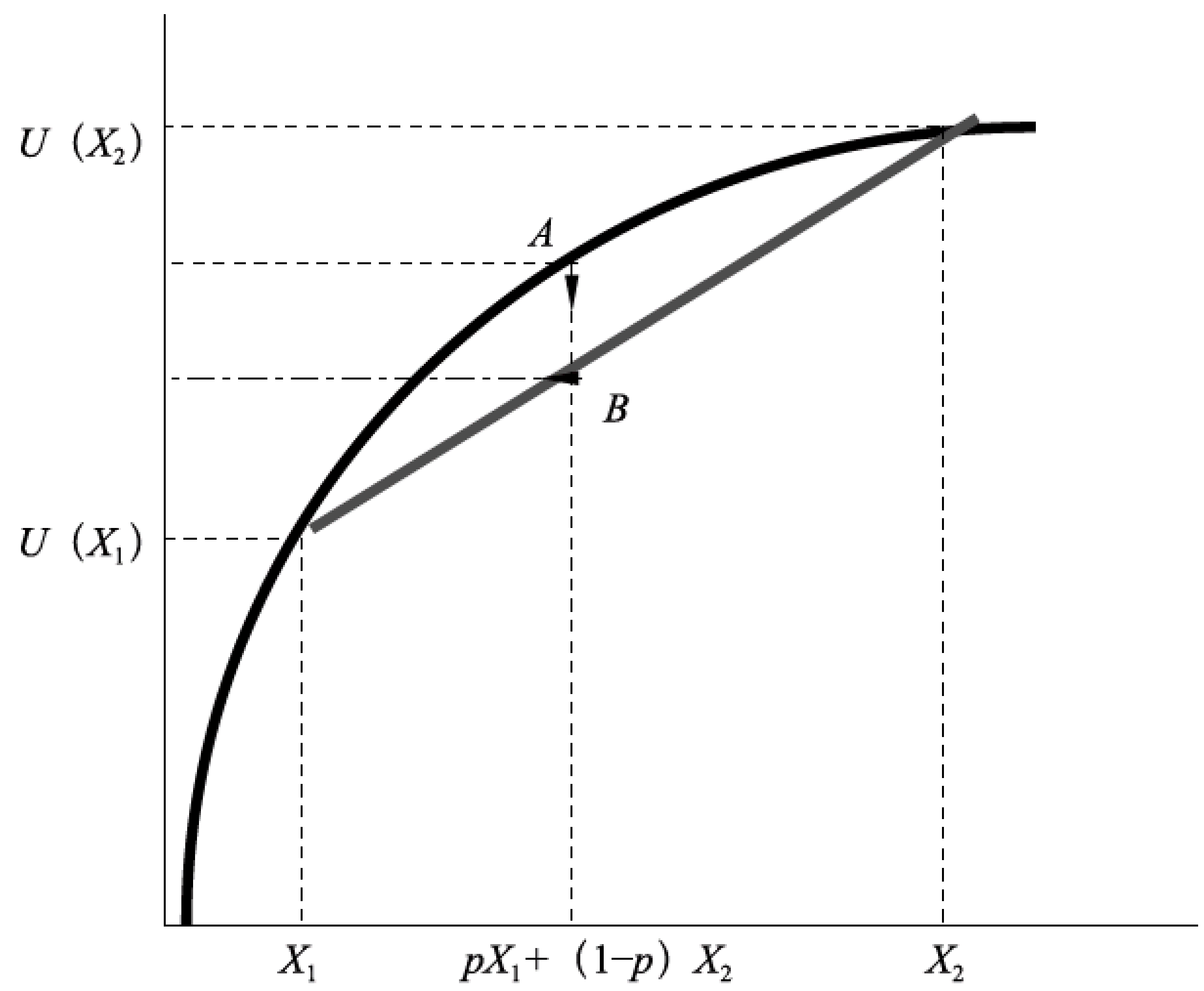


图4.1 风险厌恶下的效用曲线

2）风险寻求

假定决策者在无风险条件下所能获得的确定性收入与他在有风险条件下所能获得的期望收入相等，如果决策者对于有风险条件下期望收入的效用大于对于确定性收入的效用，则该决策者属于风险喜好型，或风险寻求者。风险寻求者的效用函数是凸函数，即其收入—效用曲线是凸向原点的。对于风险寻求者来说，货币收入所提供的总效用是以递增的速率增加，即边际效用递增。

例如，在图4.2中，A 点的期望效用为：$U[pX_1+(1-p)X_2]$，p 为0～1的常数，B 点的期望效用为：$pU(X_1)+(1-p)U(X_2)$，即 $U[pX_1+(1-p)X_2]<pU(X_1)+(1-p)U(X_2)$。该效用函数满足凸性函数的特征。

3）风险中性

假定决策者在无风险条件下的确定性收入与他在有风险条件下的等值的期望收入获得的效用是相同的，则该决策者属于风险中性型。风险中性者的收入—效用曲线是一条从原点出发的射线，该效用曲线的斜率即边际效用是既定不变的，或者说对于风险中性者来说，货币收入所提供的总效用是以不变的速率增加，即边际效用不变。例如图4.3，$U[pX_1+(1-p)X_2]=pU(X_1)+(1-p)U(X_2)$

例如，有一种风险型彩票，抽奖者有1/2的概率获得90元，1/2概率得到0元。

假定由该风险型彩票抽奖得到90元的效用值为12，得到0元的效用值为2，那么通过

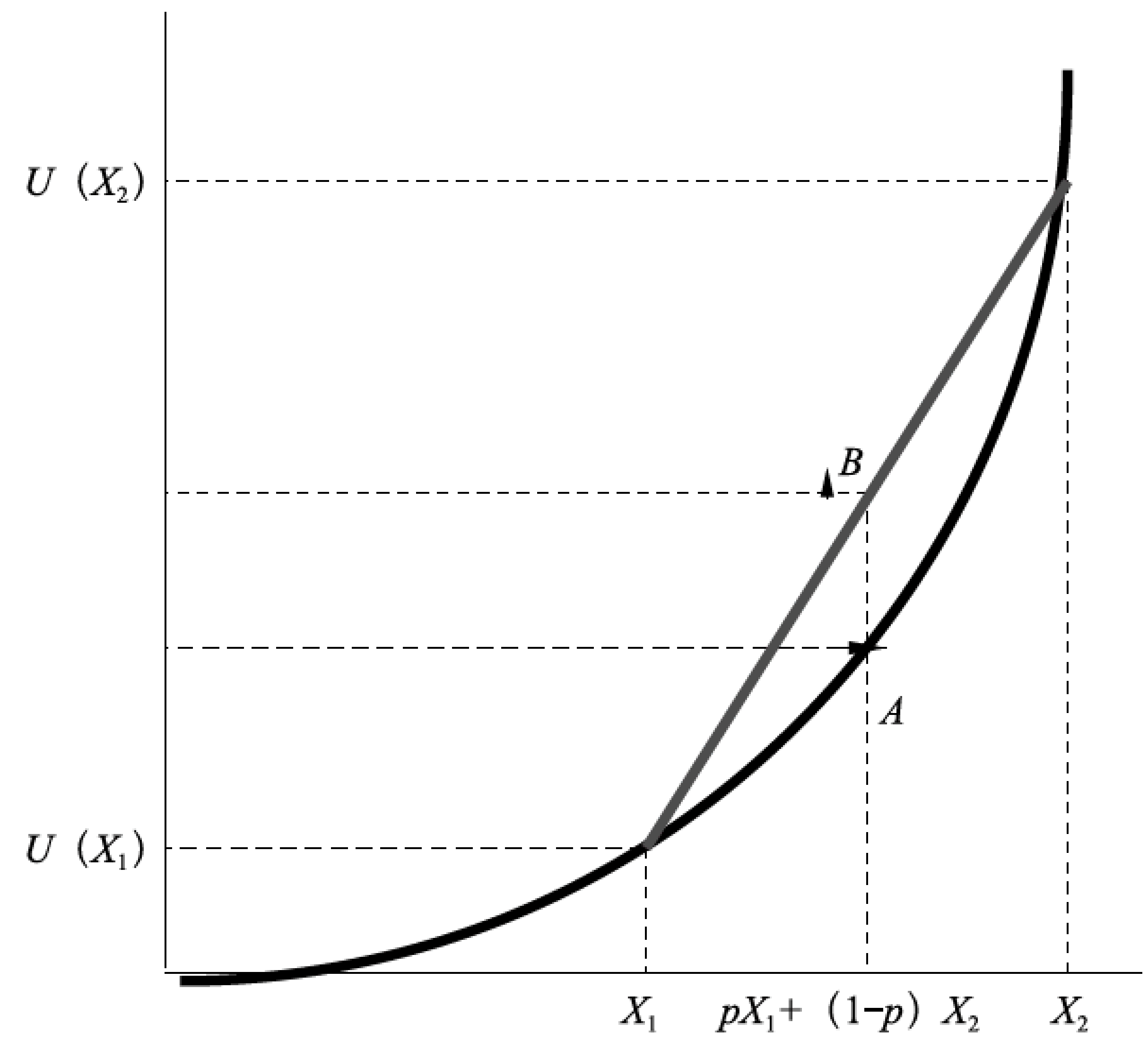


图 4.2　风险寻求下的效用曲线

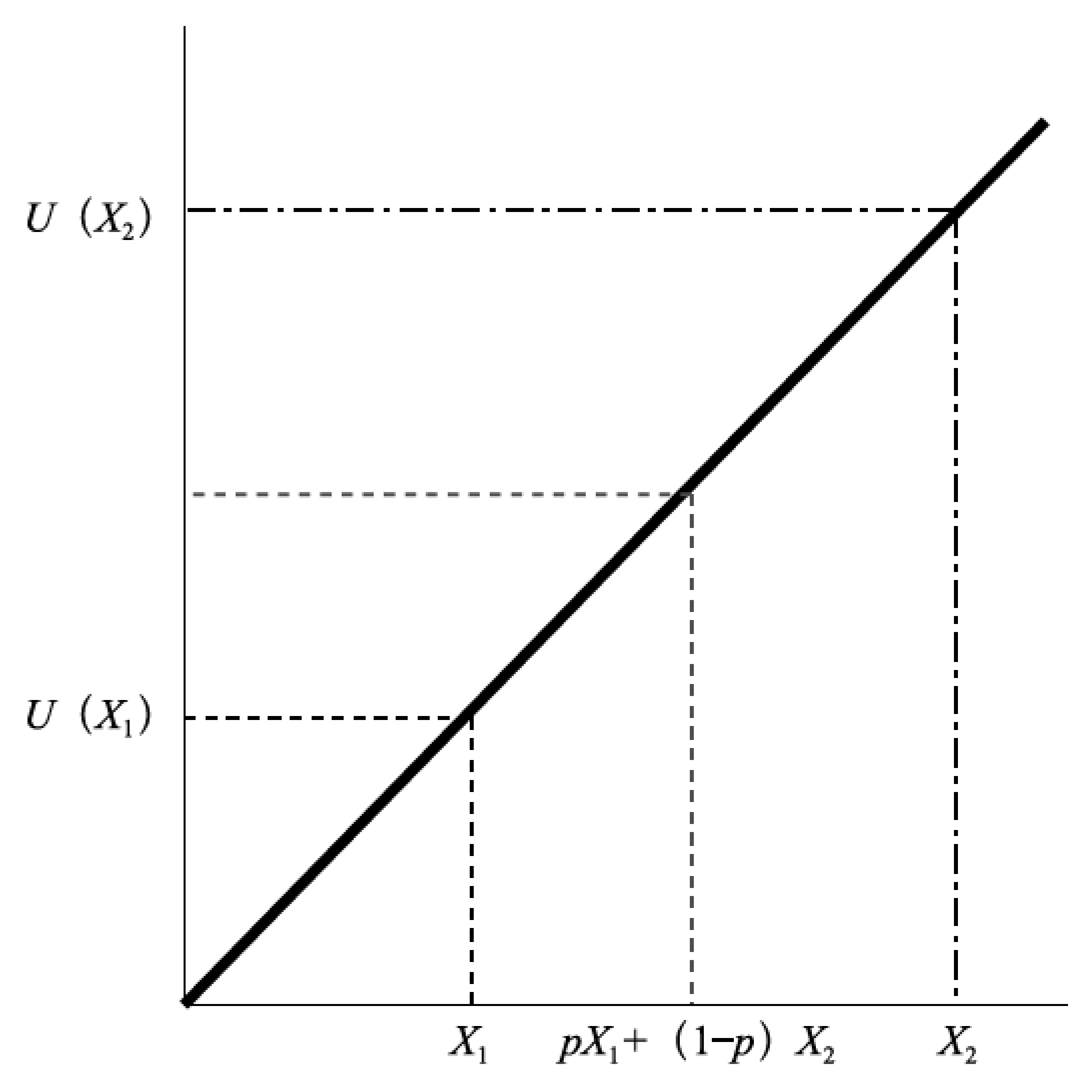


图 4.3　风险中性下的效用曲线

该彩票可以获得的期望效用值为

$$EU=1/2\times 12+1/2\times 2=7$$

在该风险型彩票中得到的货币收入的数学期望值为

$$EM=1/2\times 90+1/2\times 0=45$$

如果现在另有一种安全彩票，可以确定性地得到 45 元，请问在风险型彩票抽奖和确定性得到 45 元之间，你愿意选择哪一个？

分析：将确定性得到 45 元的效用与风险型彩票的期望效用作以下比较。

如果一个人偏好确定性的 45 元，即对他来说，$U(45)>7$，那么他是风险厌恶型的；

如果一个人偏好风险型彩票，即对他来说，$U(45) < 7$，那么他是风险寻求型的；

如果一个人对这两种彩票的偏好没有差异，即对他来说，$U(45) = 7$，那么他是风险中性的。

4. 风险态度的测量

风险态度是决策者的一个重要特征，会对其行为产生重要的影响。在研究和实践中，有很多方法用于测量个体的风险态度，包括问卷法、实验法等。例如，在中国，如果一个客户到银行购买理财产品，他通常会被要求填写一系列问卷，以供银行销售人员判断客户的风险态度。如果客户的风险态度是风险厌恶的，则银行一般不能向其推销风险较大的理财产品。如何合理、准确地评估个体的风险态度是一个非常值得深入探讨的问题。

1）实验法

由于实验法可控制性、可重复性强的特点，用它来测量个体的风险态度在文献中得到了广泛的应用。其中，最有名的要数美国弗吉尼亚大学 Holt 和 Laury 设计的实验。在这个实验中，被试被要求作出 10 个决策，每个决策都是在两个彩票之间进行选择，如表 4.1 所示。

表 4.1 Holt 和 Laury 风险态度实验

决策	选项 A	选项 B	你的选择（A 或者 B）
1	10%的机会得到 2 美元，90%的机会得到 1.6 美元	10%的机会得到 3.85 美元，90%的机会得到 0.1 美元	
2	20%的机会得到 2 美元，80%的机会得到 1.6 美元	20%的机会得到 3.85 美元，80%的机会得到 0.1 美元	
3	30%的机会得到 2 美元，70%的机会得到 1.6 美元	30%的机会得到 3.85 美元，70%的机会得到 0.1 美元	
4	40%的机会得到 2 美元，60%的机会得到 1.6 美元	40%的机会得到 3.85 美元，60%的机会得到 0.1 美元	
5	50%的机会得到 2 美元，50%的机会得到 1.6 美元	50%的机会得到 3.85 美元，50%的机会得到 0.1 美元	
6	60%的机会得到 2 美元，40%的机会得到 1.6 美元	60%的机会得到 3.85 美元，40%的机会得到 0.1 美元	
7	70%的机会得到 2 美元，30%的机会得到 1.6 美元	70%的机会得到 3.85 美元，30%的机会得到 0.1 美元	
8	80%的机会得到 2 美元，20%的机会得到 1.6 美元	80%的机会得到 3.85 美元，20%的机会得到 0.1 美元	
9	90%的机会得到 2 美元，10%的机会得到 1.6 美元	90%的机会得到 3.85 美元，10%的机会得到 0.1 美元	
10	100%的机会得到 2 美元，0%的机会得到 1.6 美元	100%的机会得到 3.85 美元，0%的机会得到 0.1 美元	

表 4.1 中，在每个决策中，选项 A 比选项 B 的方差更小，也就是更安全，所以一般把选项 A 称为安全选项，把选项 B 称为风险选项。如果进一步仔细计算，比较表中每个彩票的

期望货币价值，会发现从决策 1 到决策 4，选项 A 的期望货币价值都超过选项 B；从决策 5 开始，选项 B 超过选项 A。根据期望效用理论，对于风险中性的被试，应该在前 4 个决策中选择 A，后 6 个选项中选择 B；风险寻求的被试会在第 4 个选项之前、更早地从 A 变化到 B；风险厌恶的被试则会在第 4 个选项之后、更晚地从 A 变化到 B。简单而言，可以用被试选择选项 A 的个数代表其风险偏好程度，选择 A 的个数越多，被试越风险厌恶；反之，则越风险偏好。

除了彩票选择之外，Charness 和 Genicot(2004)设计的投资实验是请被试选择在 100 元现金中，有多少投资于风险资产，有多少投资于无风险资产。如果被试选择 X 元投资于风险资产，有 0.5 的概率 X 元增长为原来的 2.5 倍，有 0.5 的概率 X 元变为零；选择投资在无风险资产上的$(100-X)$将保持不变。被试选择的 X 数值越高，代表越高的风险偏好。

吹气球任务实验(balloon analog risk task，BART)也是测量风险偏好的一个常用方法。在该实验中，被试选择在计算机上按键，向一个虚拟的气球吹气，按键次数越多，意味着吹进更多的气，如果气球不爆炸，能获得更高的收益，如果气球爆炸，则收益为零。越是风险偏好的被试，会更多次地按键。

实验法的一个核心要点是，要在实验结束之后根据被试的选择付给被试真实的货币。在这种激励方式下，被试的选择在很大程度上能够代表他真实的偏好特征。

2) 问卷法

问卷法测量风险态度常用于广泛、大规模的调查研究。例如，Guiso 等(2007)，李涛和郭杰(2009)在问卷研究中用这样一个问题测量被调查者的风险偏好。

假如让您花钱玩一个游戏，让您在一个装有 100 个球(其中 50 个红球、50 个黑球)的罐子中随意取出一个球，如果它是红球，您可以获得 2 500 元；如果它是黑球，您将一无所得。您最多愿意花多少钱玩这个游戏？

被调查者填写的金额越高，代表其具有更高的风险偏好。

Weber 等(2002)设计了一个量表，通过六大类型的 40 个问题来综合衡量个体的风险偏好。这些类型包括社会问题、投资问题、伦理问题、健康问题、赌局问题、娱乐问题等。例如，你是否会要求老板涨工资(社会问题)；你是否会用一天的收入去赌马(投资问题)；你是否会偷税漏税(伦理问题)；你是否坐在汽车前排时不系安全带(安全问题)等。被调查者要求在代表“完全不同意”……“完全同意”的 1～5 点中进行选择。

专栏 4.1 风险态度与留学决策

风险态度对人们在日常生活中的选择有很大的影响，如有研究发现越是风险厌恶的人，越不愿意到外地去就业。我们的一个研究利用浙江大学学生风险态度的实验数据和他们毕业后的升学数据，检验了风险态度对是否出国留学产生的影响。我们发现，选择出国留学的学生比选择在国内就业和升学的学生有更高的风险偏好水平；相对于到亚洲和美国留学的学生而言，选择到欧洲留学的学生更加偏好风险。在控制了家庭收入、在校学习成绩之后，风险态度对是否选择留学有着显著的影响。

资料来源：杨晓兰. *Chinese college students' risk attitude to moving abroad to study*[J]. *Social behavior and personality*, 2015, 43(5).

4.2 心理学实验对期望效用理论的挑战

期望效用理论是经典的在不确定环境下的决策模型,其假设基础是“理性人”,而非真实的现实人,该理论在预测与解释真实行为时得到了一些支持,但是,也有大量的证据证明个体在决策中并不是最大化他们的期望效用,期望效用理论不能解释很多观察到的行为。正如特沃斯基和卡尼曼(1986)所提出的,期望效应理论与真实行为的偏离是“如此的广泛而不能被忽视,是如此的系统性而不能将其视为随机误差,是如此的基础而不能通过放松标准模型的条件而容纳进去”,因此,学者纷纷通过各种方式对期望效用理论进行检验。

期望效用理论受到的主要挑战有:期望效用理论难以解释阿莱悖论、Ellsberg 悖论等现象;没有考虑现实生活中个体效用的模糊性、主观概率的模糊性;不能解释偏好的不一致性、非传递性、不可替代性、偏好逆转现象等;现实生活中也有对期望效用理论中理性选择上的占优性原则和恒定性原则的违背;实际生活中的决策者对效用函数的估计也违背期望效用理论的效用函数。另外,随着实验经济学的发展,期望效用理论在实验经济学的一系列选择实验中受到了一些“悖论”的挑战。实验经济学在风险决策领域所进行的实验研究最广泛采取的是彩票选择实验(lottery-choice experiments),即要求被试在一些配对的组合中进行选择,这些配对的选择通常在收益值及赢得收益值的概率方面存在关联。通过实验经济学的论证,确定性效应、同结果效应、同比率效应、反射效应、孤立效应、偏好逆转等“悖论”的提出对期望效用理论形成了巨大冲击。

1. 确定性效应、同结果效应和同比率效应

1)确定性效应

最早对期望效用理论提出质疑的实验来自法国经济学家阿莱(1953),他提出了“阿莱悖论”(Allais paradox,1953),他的实验结果引发了无数相关的研究。阿莱悖论是在不确定性条件下选择行为违背期望效应理论的典型例子,这个悖论中利用了确定性效应(certainty effect)。确定性效应是指相对于不确定性的收益,人们赋予确定性的收益更高的权重,或者说过度重视确定性结果。这一效应表明,在特定情况下,人们的效用函数会低估一些只是可能性的结果,而相对高估确定性的结果,导致决策者在面临条件相当的收益期望时,更倾向于接受确定性的收益结果。

实验 4-1

“阿莱悖论”

实验设计:实验者被要求分别在下面两组彩票中作出选择。

问题 1:A. 有 100%的机会得到 100 万美元,即($100 万,1.00)。

B. 有 10%的机会得到 500 万美元,89%的机会得到 100 万美元,1%的机会什么也得不到,即($500 万,0.10;$100 万,0.89;$0,0.01)。

问题 2:C. 有 11%的机会得到 100 万美元,89%的机会什么也得不到,即($100 万,0.11;$0,0.89)。

D. 有10%的机会得到500万美元,90%的机会什么也得不到,即($500万,0.10;$0,0.90)。

实验结果:多数人选择了(A,D)。

实验结果分析:

在阿莱的实验中,被试被要求对两组彩票进行选择,根据期望效用理论,这两组彩票A、B与C、D的偏好关系应该是完全一致的,即如果人们偏好A,那么也会偏好C;如果偏好B,也会偏好D,因此与期望效用理论一致的选择是(A,C),或者(B,D),这是由期望效用理论的独立性假设推导出来的;违背期望效用理论的选择是(A,D)或者(B,C)。

用效用函数U来代表实验者的效用,实验结果表明,在问题1中,多数人偏好A而不是B,这就意味着:

$U(1)>0.01U(0)+0.89U(1)+0.10U(5)$,也就是$0.11U(1)>0.01U(0)+0.10U(5)$……(1);

在问题2中,多数人偏好D而不是C,这就意味着:

$0.90U(0)+0.10U(5)>0.89U(0)+0.11U(1)$,也就是$0.01U(0)+0.10U(5)>0.11U(1)$……(2)。

显然,式(1)和式(2)是明显矛盾的,它系统性地背离了期望效用理论,至少违背了期望效用理论关于偏好的传递性以及恒定性等公理性假定。实际上,问题2中的C和D分别是由问题1中的A和B分别减去“89%的机会得到100万美元”($100万,0.89)而得到的,因此与期望效用理论相一致的选择是实验结果(A,C),或者(B,D),但是实验结果中大部分人选择了(A,D)。在问题1中,A中的100万美元是确定的,因此更吸引人;但是在问题2中,100万美元不再是确定的,这种变化使得期望的性质从确定性收益变成了可能性收益,从而导致人们的预期发生较大幅度的改变。这一实验结果由阿莱1953年首次发现,被称为阿莱悖论,选择(A,D)的行为被称为阿莱行为(Allais behavior)。

专栏4.2 消失的“阿莱悖论”

有实验发现阿莱悖论消失的现象。Conlisk(1989)以236个加利福尼亚大学的本科生组成被试,对实验4-1的彩票进行了选择实验,在支付是假设性时,有50.4%的被试(有119人)选择了(A,D)或者是(B,C),并且多数人选择(A,D),实验结果((A,C):18人;(B,D):99人;(A,D):103人;(B,C):16人)。当Conlisk把支付变成真实的,并且把奖金金额降低至($0.5,$25),实验结果显示人们在两个问题中都选择了货币期望值高的那个选项,即问题1选B,问题2选D,阿莱行为在这种情况下消失了。

编者多年来一直在《行为金融》课程的教学中,以选课学生(浙江大学,本科,三年级)作为被试做上述彩票选择实验,在支付是假设的情况下,实验结果没有发现阿莱悖论(A,D),实验中绝大多数人选择(B,D)。例如,2018年3月的实验结果为:第一组(72人),87%的被试在问题1中选择B,95%的被试在问题2中选择D。第二组(53人),77%的被试在问题1中选择B,88%的被试在问题2中选择D。当然,为什么阿莱悖论会消失,背后可能的原因值得我们思考。

2）同结果效应

同结果效应是指对相同结果导致不一致偏好的情形。确定性效应采用的是加法变换，最早通过阿莱悖论所反映，它对期望效用理论形成了挑战。在阿莱的实验之后，又有许多学者进行大量重复实验，结果也都发现了该效应的存在。

实验 4-2

同结果效应

实验设计：实验者被要求分别在下面两组彩票中作出选择。

问题 3：A：（＄2 500，0.33；＄2 400，0.66；＄0，0.01）。

B：（＄2 400）。

问题 4：C：（＄2 500，0.33；＄0，0.67）。

D：（＄2 400，0.34；＄0，0.66）。

实验结果：绝大多数人选择了（B，C）。

实验结果分析：实验结果显示，多数人在问题 3 中选择 B，在问题 4 中选择 C。实际上问题 4 是由问题 3 同时取消（＄2 400，0.66）得来，与期望效用理论相一致的选择是实验结果（A，C），或者（B，D）。实验结果表明大部分人选择了（B，C），在两个问题中都作出了情绪化选择，所反映的是对相同结果的不一致偏好情形，即“同结果效应”。此实验违背了期望效用理论关于偏好的传递性、占优性以及恒定性（独立性）等公理化假设。

3）同比率效应

与同结果效应类似的实验发现是同比率效应（common ratio effect）。同比率效应是指如果对一组收益概率进行相同比例的变换，也会产生不一致的选择，采用的是乘法变换。它不仅存在于货币结果的彩票对选择中，也存在于非货币结果的选择决策中。这一效用最早由卡尼曼和特沃斯基（1979）提出。

同比率效应违反了偏好的消除性假设，其公式化表达如下：

彩票组（$b1$，$b2$）与（$b3$，$b4$），其中：

$b1=(\$X,p;\$0,1-p)$；

$b2=(\$Y,q;\$0,1-q)$；

$b3=(\$X,rp;\$0,1-rp)$；

$b4=(\$Y,rq;\$0,1-rq)$；

其中满足下列条件：$p>q$，$0<X<Y$，$0<r<1$。

实验 4-3

同比率效应

实验设计：在卡尼曼和特沃斯基（1979）的例子中 $p=1$，实验者被要求分别在下面两组彩票中作出选择。

问题 5：A.（＄4 000，0.80；＄0，0.20）。

B.（＄3 000）。

问题 6：C.（＄4 000，0.20；＄0，0.80）。

D.（3 000，0.25；＄0，0.75）。

实验结果：被试数 $N=95$，问题 5：A(20％)，B(80％)；问题 6：C(65％)，D(35％)。

实验结果分析：有 80％的被试选择了 B，而有 65％的被试选择了 C。如果令 $U(0)=0$，则选择 B 意味着 $U(3\ 000)/U(4\ 000)>4/5$，而选择 C 则意味着相反的不等式。实际上问题 6 是由问题 5 演化而来的，只不过赢的概率减少了 3/4，即 C 可以表示为(A，0.25)，而 D 可以表示为(B，0.25)。如果按照期望效用理论，问题 5 中的 B 偏好于 A 则意味着问题 6 中的 D 偏好于 C。这说明问题 3 和问题 4 中偏好的常见模式与期望效用理论不符，当赢的概率大幅度降低后，大多数人的偏好从较低的奖赏转移到了较高的奖赏。把赢的概率从 1 降低到 0.25 所产生的影响比从 0.8 降低到 0.2 所带来的影响更大。

编者多年来以《行为金融学》课程选课学生作为被试做上述彩票选择实验，实验结果和卡尼曼和特沃斯基(1979)的结果一致，出现同比率效应，绝大多数人在实验中选择(B，C)。例如，2018 年 3 月的实验结果为：第一组(72 人)，61％的被试在问题 5 中选择 B，75％的被试在问题 6 中选择 C。第二组(53 人)，69％的被试在问题 5 中选择 B；75％的被试在问题 6 中选择 C。

除了货币收益的实验外，卡尼曼和特沃斯基(1986)对非货币收益的实验也得出了相同结论。他们对参加加利福尼亚医学会会议的 72 名外科大夫及 180 名大学生关于肿瘤治疗方案的测试也获得了基本一致的结果。实验设计如下：

假设对一种肿瘤疾病的治疗有两种方法可以选择。

(1) 激进疗法，如采用广泛的手术，可能导致病人很快死亡。

(2) 温和疗法，如采用有限的手术和放射疗法。

下面的例子描述了两种疗法的各种可能性结果。在每个例子中，假定患者是一位 40 岁的男子，若不进行治疗，就会很快地死亡(1 个月内)，而且只能对他实行一种治疗措施。请在每种情形中选择你喜欢的治疗措施。

情形 1：

措施 A——20％的概率会很快死亡，80％的概率预计还能正常生活 30 年。

措施 B——预计还能正常生活 18 年。

情形 2：

措施 C——80％的概率会很快死亡，20％的概率预计还能正常生活 30 年。

措施 D——75％的概率会很快死亡，25％的概率预计还能正常生活 18 年。

实验结果：被试数 $N=72$，情形 1：A(35％)，B(65％)；情形 2：C(68％)，D(32％)。

实验结果分析：在情形 1 中，有 65％的被试选择了措施 B，即作出了风险规避的决定，选择了虽然较短但确定的生存时间；而在情形 2 中，有 68％的被试选择了措施 C，即在温和疗法并不能确保患者能生存下来的情况下，多数被试选择了有较长存活时间的疗法。体现了同比率效应。

2. 反射效应

前面进行的选择性实验都是面对收益的，如果面对损失会出现什么结果呢？卡尼曼和特沃斯基(1979)发现，确定性效应不仅表现在正彩票之间的选择上，也表现在负彩票之间的选择上，他们设计了负彩票实验，实验设计如下。

实验 4-4

反射效应

实验设计

问题 5′:A(−4 000,0.80)

B(−3 000)

问题 6′:C(−4 000,0.20)

D(−3 000,0.25)

实验结果:被试者数 N=95,问题 5′: A[92%],B [8%];问题 6′:C[42%],D[58%]

实验结果分析:实验 4-3 的问题 5 和问题 6 显示出与不确定性收益相比,人们高估确定能获得的收益。本实验的问题 5′与问题 6′显示出,人们高估确定会发生的损失,低估不确定性的损失。所以,在收益区域内,确定性效应导致风险规避行为,人们偏好确定性收益而不是不确定的收益;而在损失区域,人们偏好不确定的损失而不是确定性损失。因此,过分高估确定性导致人们在损失区域的风险寻求和在收益区域的风险规避。负彩票之间的偏好和正彩票之间的偏好形成了一个镜像关系。这种以零为中心的彩票反转使得偏好顺序发生了反转,被卡尼曼和特沃斯基(1979)称为“反射效应”(reflection effect)。

“反射效应”是指个人对于获得和损失的偏好不是对称的,面对可能损失的期望时,个人有风险寻求的倾向;面对盈利期望时,个人有风险规避的倾向。确定性效应认为人们存在对不确定性的厌恶。但反射效应表明,这一结论需要加以修正,因为这种对不确定性的厌恶只是在面对收益的时候才表现出来,而面对损失的时候刚好相反。由此可见,个人选择的依据并不是最终的期望效用,而是相对于参考点的财富变动。

编者多年来以《行为金融学》课程选课学生作为被试对下述彩票进行选择实验,实验设计如下:

问题 7:A. 100%能赢得 1 000 元。

B. 50%的可能赢得 2 000 元;50%的可能什么都得不到。

问题 8:C. 100%要损失 1 000 元。

D. 50%的可能要损失 2 000 元;50%的可能什么都不损失。

实验结果与卡尼曼和特沃斯基(1979)的结果一致,出现反射效应,绝大多数人在实验中选择(A,D)。例如,2018 年 3 月的实验结果为:第一组(72 人),77%的被试在问题 7 中选择 A,79%的被试在问题 8 中选择 D。第二组(53 人),75%的被试在问题 7 中选择 A,79%的被试在问题 8 中选择 D。

3. 孤立效应

孤立效应(isolation effect),为简化在不同选项之间的选择,人们往往忽略各种选项共同的部分而集中关注不同的部分。这一选择问题的方式会引起不一致的偏好,因为当一组预期用不同方法分解成相同与不同的部分时,不同分解方式可能导致不同偏好。人们在分析评估不同的“待选择期望”时,经常暂时剔除期望中的相同因子,但是通常情况下,一组期望可以用不止一种方法被分解成相同和不同的因子,这种分解方式的多样性会导致人的偏好和选择的不一致性,卡尼曼和特沃斯基把这种现象称为“孤立效应”。个人会

因为问题描述方式的不同而有不同的分解方式和不同的选择，这也是我们在第 2 章提到的框定依赖现象。孤立效应推翻了期望效用理论中效用仅与事件的最后状态（概率分布与时间结果）有关的结论。

实验 4-5

孤 立 效 应

实验设计：实验者被要求分别在下面两组彩票中作出选择。

问题 9：C.（\$4 000，0.20；\$0，0.80）。

D.（\$3 000，0.25；\$0，0.75）。

问题 10：在两阶段的博弈中，第一阶段中有 75%的概率赢得 0 美元，25%的概率转到第二阶段；在第二阶段面临两种选择：A（\$3 000），B（\$4 000，0.80；\$0，0.20）。

要求被试必须在博弈开始之前就作出选择。

实验结果：

问题 9：被试数 $N=95$，C(65%)，D(35%)。

实验结果：被试数 $N=141$，A(78%)，B(22%)。

实验结果分析：经过计算可知，问题 10 就是要求被试在以 $25\%\times1.0=0.25$ 的概率获得 3 000 美元，和以 $25\%\times0.80=0.20$ 的概率获得 4 000 美元之间进行选择，因此从最终收益来看，其与问题 9 实际上是一致的。然而被试选择的结果却是大相径庭，在问题 10 中多数被试作出了貌似风险规避的选择，而问题 9 中却没有这样的选择结果。这种现象就是孤立效应，也被称为“伪确定性效应”。如果用树状图将问题 9 和问题 10 表述出来（图 4.4 和图 4.5），会更加直观。

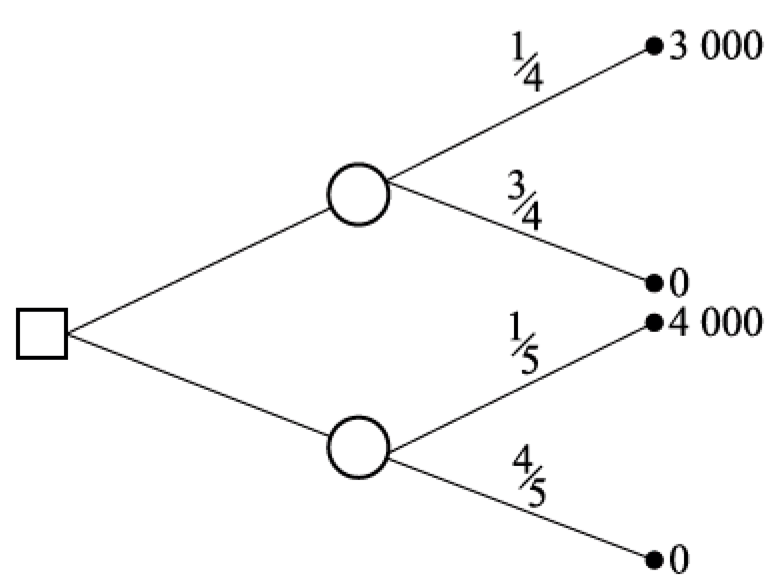


图 4.4 问题 9

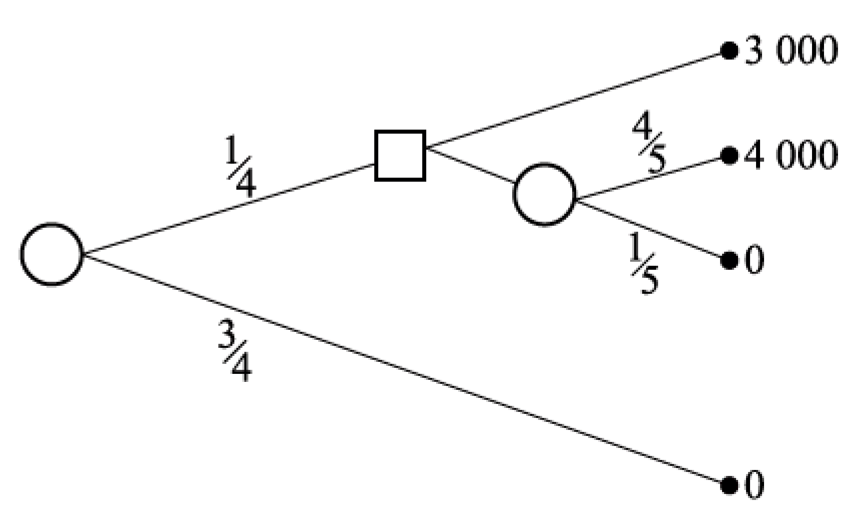


图 4.5 问题 10

4. *偏好逆转*

可传递性是经典经济学偏好的基本属性，然而一系列实验的结果未能支持这个属性，偏好逆转现象就反映了不可传递性。所谓偏好逆转，就是个体决策与偏好方面选择与定价不一致的现象。经典的偏好逆转实验是由 Lichtenstein 和 Slovic(1971)设计的，之后迅速获得了广泛的证实，并掀起了一场声势浩大的研究热潮。如果说同结果效应和同比率效用等实验发现对期望效用理论提出的尚属挑战的话，那么偏好逆转现象的实验发现则对期望效用理论形成了真正的冲击。

最早发现偏好逆转现象的是 Lichtenstein 和 Slovic(1971)，他们对发现过程说明如下：“促使我们进行这项研究的动力是我们 1968 年发表的文章中的观察结果：在成对彩票

之间的选择似乎主要受输赢的概率影响，而买卖价格却主要是受输赢的货币数额影响……让被试对一个有吸引力的彩票确定一个价格时，他们好像是从可以赢得的数额开始考虑的，进而往下调整，同时考虑输赢的概率和可能输的数额。这种调整过程较含糊，让价格反映在很大程度上受起始点价格的影响。另一方面，选择好像受不同规则的支配。我们设想，如果进行选择和确定价格时对信息的处理不同，那么就应该能构建出一组成对的彩票，使得人们可以从彩票中选取其中的一种，但对另一种彩票定出较高的价格……”于是，他们构建了一组成对彩票实验，来验证这种预期的结果。

实验 4-6

机会彩票与金钱彩票

实验设计：

(1) 被试被要求在以下形式的彩票中作出选择。

A. 机会彩票：表示赢的机会(概率)较大、但可能赢得金额较小的彩票；用简约形式表示为：$(p, X; 1-p, x)$。

B. 金钱彩票：表示赢的金额较大、但赢的概率较小的彩票；用简约形式表示为：$(q, Y; 1-q, y)$。

其中满足：$X>x, Y>y, p>q, Y>X$。

(2) 让被试在下列几种情况下对这些彩票作出评价，说出其确定性等值。

A. 让被试指出，如果可以转让参加这些彩票的权利，对于每种彩票，他们愿意接受的最低卖价或支付意愿是多少。

B. 如果让被试买下，则请指出他们的最高买价或接受意愿。

C. 采用对说“真情”者给予实质性奖励的方法，力图让被试准确地说出彩票的最低出让价格。

实验结果分析：如果按照期望效用理论，被试应该选择确定性等值较高的彩票；但实验结果却发现了一种背离倾向：被试一方面选择了机会彩票，而另一方面却对金钱彩票评价较高。在每一次彩票中均选择机会彩票的173个被试中，有127名被试对金钱彩票给出了较高的买价。

Lichtenstein 和 Slovic(1971)的实验表明，具有同一偏好的同一个人对同一彩票集合进行选择时，却表现出相反的两种风险偏好，这种不一致的行为选择就是“偏好逆转”现象。偏好逆转现象中的一个重要发现是“接受意愿”与“支付意愿”间的巨大悬殊，也就是如果让人们对某种经济利益进行定价，则其得到这种经济利益所愿意支付的最大值，远远小于其放弃这种经济利益所愿意接受的最小补偿值。除 Lichtenstein 和 Slovic 的实验之外，Lindman(1971)、Grether 和 Plott(1979)的实验，以及 Lichtenstein 和 Slovic(1973)在拉斯维加斯赌场中的实验都得到了同样的结果。

实验 4-7

Grether 和 Plott(1979)实验

Grether 和 Plott(1979)发现早期关于偏好逆转实验中存在的一些问题，主要是没有使用真实的货币支付，或者即使使用了真实的货币支付，但缺少对财富效用的控制等。因

此，在他们的实验中尽可能对各种实验条件进行控制与测试，如采用真实激励、控制财富效应、排除收入效应与替代效应、检测策略性反映即讨价还价效应，甚至采用不同的概率器等，但仍然发现偏好逆转现象普遍存在。

他们将被试分成两组，其中一组采用假设性选择，而另一组采用给予显著报酬的真实选择。在两组实验中均提供给被试6对成对的彩票，形成12个彩票，在此基础上要求被试在每对彩票中作出选择，并对12个彩票给出最低愿意出让的价格。实验者从所有决策中随机选择1个来确定货币支付额，从而能较好地控制财富效应和组合效应。为了保证有正的净收益，在每局开始赋予每个被试初始禀赋7美元。他们的实验结果如表4.2所示。

表4.2 Grether 和 Plott（1979）偏好逆转实验结果

组别	选择机会彩票的被试所占的比例/%	选择机会彩票而发生偏好逆转的比例/%	选择金钱彩票而发生偏好逆转的比例/%
假设性选择组	49	56	11
真实货币支付组	36	70	13

从表4.2中可以发现两点：①发生偏好逆转的概率与彩票的选择有很大相关性，选择机会彩票发生偏好逆转的可能性大很多；②如果在实验中引入真实的货币激励，偏好逆转的发生频率将会显著增加，特别对机会彩票而言更为明显。

对于偏好逆转的讨论十分广泛，尽管对于其产生的原因及机制方面存在不同的看法与解释，基本上认为偏好逆转违背了期望效用理论的传递性假设。正如特沃斯基和塞勒（1990）所说的："它几乎违背了经济学中关于偏好的所有原则"；而 Grether 和 Plott（1979）则悲观地声称："偏好逆转现象让人觉得哪怕是最简单的人类选择行为，都不存在任何种类的最优化法则。"

偏好反转的现象说明，人们并不拥有事先定义好的、连续稳定的偏好，偏好是在判断和选择的过程中体现出来的，并受判断和选择的背景、程序影响。

5. 非明显占优

特沃斯基和卡尼曼（1986）的非明显占优（nontransparent dominance）实验，形象地说明了占优关系是否会被掩盖的问题。在前景理论里，特殊地，对于小概率 p，π 函数是次可加的（subadditive），如 $\pi(0.01)+\pi(0.06)>\pi(0.07)$。这个性质往往会导致占优性的偏离，如下面的实验所示。

问题11：（被试数 $N=88$）考虑下面两个彩票，参与者赢或输的金额由盒子中随机抽取的球的颜色决定，请问你会选择下面哪个彩票？

	90%白	6%红	1%绿	1%蓝	2%黄
A.	0$	赢$45	赢$30	输$15	输$15
B.	0$	赢$45	赢$45	输$10	输$15

很明显B彩票要比A彩票好，即B彩票是占优的，对于每一种颜色，B彩票的结果至少和A彩票的结果一样好。事实上，所有的被试都选择了B彩票，而不是A彩票，这个结

果并不令人惊奇，因为其中的占优关系很明显，不需要进一步地思考。

但如果将这个问题中产生相同结果的颜色进行合并，即在A彩票中将蓝色和黄色进行合并、B彩票中将红色和绿色进行合并，被试的选择将发生改变，如下。

问题12：（被试数 $N=124$）请问你会选择哪个彩票？

C.	90%白	6%红	1%绿	3%黄
	0$	赢$45	赢$30	输$15
D.	90%白	7%红	1%蓝	2%黄
	0$	赢$45	输$10	输$15

实际上，问题12是对问题11的简化，但却掩盖了占优关系。看起来，似乎C彩票显得更有吸引力：有2个正结果和1个负结果；而D彩票却只有1个正结果和2个负结果。为了诱导参与者能更仔细地思考各彩票，主试告诉随机选择的10%的参与者，将按照其选择进行实际博彩。尽管引起了较大的震动，但是仍然有58%的被试选择了劣势的C彩票；在回答另一个问题时，大多数被试仍然对C彩票比D彩票给予了更高的现金评价。

上述结果支持了下述性质：①同一问题的两种不同表达引起了不同的偏好，导致了对恒定性的偏离；②当占优性很明显时，占优规则将得到遵守；③占优性可由框定掩盖，在实际相同的状态下，较劣势选择将产生更有利的结果（这里的抽出绿色球）；④偏好的差别与决策权重的次可加性一致。

阿莱悖论、确定性效应、同结果效应、同比率效应和偏好逆转等现象表明，人们在风险和不确定条件下进行选择时，系统性地违背了期望效用理论的公理性假设，这些发现对以期望效用理论为核心的现代决策理论形成了巨大冲击，并进一步向经济学的许多理论命题提出了严峻挑战。这促使经济学界通过放松个体决策与偏好的有关公理性假设等途径，对期望效用理论模型进行修正。

4.3 前景理论

许多实验经济学家和实验心理学家从不同侧面提出了对期望效用理论的修正或扩展模型，这些修正模型主要有四类：扩展性效用模型（generalized utility model）、预期比率模型（expectation utility model）、非传递性效用模型（non-transitivity utility model）和非可加性效用模型（non-additivity utility model）。但是，从总体上说，这些修正模型并不十分令人满意。因为对某些公理化假定的放松或进行技术上的修补，只是让现象适应理论，而不能使理论解释现象。另外，这些模型本身在进一步的实验面前也经不住检验。

卡尼曼和特沃斯基提出的前景理论在一定程度上对个体决策与偏好的实验结果提供了合适的解释，是对期望效用理论的某种替代，影响也最大。

所谓前景（prospect），就是各种风险结果，前景选择所遵循的是特殊的心理过程和规律，而不是期望效用理论所假设的各种公理。卡尼曼和特沃斯基定义一个前景是一个不确定事件 $(x,p;y,q)$，个人得到 x 的概率为 p，得到 y 的概率为 q，另外 $1-p-q$ 的概率得不到任何东西。

1. 个人风险决策过程

卡尼曼和特沃斯基(1979)认为,个人在风险条件下的选择过程可以分为两个阶段:编辑阶段(editing)和估值阶段(evaluation)。编辑阶段是对所提供的彩票进行初步分析,对相关的收益和概率进行变形处理,找出其更为简便的表现形式,使决策任务变得容易;估值阶段是对编辑过的彩票进行评价,并选择出价值最高的。

1) 编辑阶段

编辑阶段的作用是对选项进行重新组织,以简化随后的估值和选择。这一阶段主要包括编码(coding)、合并(combination)、剥离(segregation)、删除(cancellation)、简化(simplification)和占优检查(detection of dominance)。

在编码阶段,卡尼曼和特沃斯基(1979)提出,人们通常关注的是收益和损失,而不是财富或福利的最终状态。而收益和损失是相对于某一参考点而言的,参考点的选择与现有的资产状况有关,或者参考点即为现有资产状况。编码就是根据参考点,对现有财富的实际收入和支出,把期望行为组合编译成决策者自己的获利或损失。例如,一个抛硬币的彩票,若是人头面朝上,你可以赢得 10 元;若是字面朝上,你将输掉 10 元。编码:(10,0.5;−10,0.5)。参考点的位置以及对收益或损失的编码,会受彩票的表达方式和决策者预期的影响。

合并是指把一致结果的概率进行合并,简化某些彩票。例如,可以将(200,0.35;200,0.25;0,0.40)简化为(200,0.60;0,0.40)。

剥离是将某些彩票中的无风险部分从有风险部分中分离出来。例如,(300,0.60;200,0.40)可以分解为确定性收益 200 和风险收益(100,0.60;0,0.4)。

删除是人们在选择中撇开了彩票中的共同部分。隔离效应表明,人们在选择中抛开了期望中共有的部分,被试忽略了第二阶段博弈中的第一阶段,因为这一阶段是两个选项中共有的,他们只考虑第二阶段的结果来对期望估价。另一种情况是人们往往抛弃共有的组成部分。如在(200,0.20;100,0.50;−50,0.30)和在(200,0.20;150,0.50;−100,0.30)之间选择,可以将共同的以 0.20 的概率赢 200 元删除,简化为(100,0.50;−50,0.30)和(150,0.50;−100,0.30)。

简化是指通过约略概率或结果从而对彩票进行简化。例如,(101,0.49)很可能被简化成(100,0.5),简化的形式包括审视彩票以发现起控制作用的选择因素,没有进一步的估价。

占优检查是检查所有给定的期望选项,以删去那些被另一个选项完全占优的选项,简化决策。

编辑工作不仅能简化估价,而且有时候能影响决策者的决策。

2) 估值阶段

在编辑阶段之后,决策者对每一个被编辑过的 Prospect 加以估值,并作出选择,选择出最好的 Prospect。

卡尼曼和特沃斯基改变了传统理论评估总效用的做法,而是提出了一个衡量 Prospect 总价值的变量 V,该价值是各价值 v 与权重 π 的加权线性之和。v 反映了结果的主观价值,与传统的效用函数 U 度量结果的最终财富不一样的是,v 衡量的是该结果离开参考点

的程度，也就是收益或者损失。π 表示与结果概率 p 相对应的决策权重，它和客观概率 p 有着本质的区别，它反映了 p 对整个 Prospect 价值的影响力。

在估值阶段中，最主要的是价值函数 v(value function)和权重函数 π(weighting function)。价值函数 v 反映的是结果的主观价值，分配给每一结果 x 一个价值量 $v(x)$，结果的定义与参考点相对应，在价值尺度中以0为参考点，因此 v 度量离开参考点的价值程度，即收益或损失；权重函数 π 表示与概率 p 相对应的决策权重 $\pi(p)$，它反映确定的概率 p 对全部价值的影响力。

当前的判断与 Prospect 的简化形式 $(x,p;y,q)$ 有关，在这个 Prospect 中，一个人以概率 p 获得 x；以概率 q 获得 y；以概率 $1-p-q$ 获得0，这里 $p+q\leqslant 1$。当结果绝对为正时(都是肯定的)，被提出的 Prospect 就是严格为正的，即 $x,y>0$，且 $p+q=1$；当结果绝对为负时(都是否定的)，被提出的 Prospect 就是严格为负的，即 $x,y<0$，且 $p+q=1$；而如果 $p+q<1$，同时 $x\geqslant 0\geqslant y$ 或者 $x\leqslant 0\leqslant y$，这个 Prospect 就属于常态的。

根据前景理论，其价值模型的基本方程式的表达分为两种情况。

第一，如果 Prospect 是常态的，也就是 $p+q<1$ 或者 $x\geqslant 0\geqslant y$，或 $y\geqslant 0\geqslant x$，则 Prospect 的价值为

$$V(x,p;y,q)=\pi(p)v(x)+\pi(q)v(y) \tag{4-1}$$

其中，V 为结果的期望值；π 为决策权重函数；$v(x)$ 和 $v(y)$ 分别是 Prospect 不同结局的价值。

第二，如果 Prospect 是严格为正或者严格为负的，人们往往先将其分解成两个部分：一是无风险部分，即可以确定获得的最小收益或者确定支付的最小损失；二是风险部分，即可能发生的收益或损失。这种 Prospect 的评价方式表达为如下。

假设 $p+q=1$ 且 $x>y>0$ 或 $x<y<0$，那么其价值为

$$V(x,p;y,q)=v(y)+\pi(p)[v(x)-v(y)] \tag{4-2}$$

也就是说，严格为正的 Prospect 和严格为负的 Prospect 的价值等于无风险部分的价值 $v(y)$ 加上不同结果之间的价值差 $[v(x)-v(y)]$ 乘上比较极端的结局的相关权重 $\pi(p)$。例如 $V(300,0.25;100,0.75)=v(100)+\pi(0.25)[v(300)-v(100)]$。从式(4-2)可以看出无风险部分的价值是 $v(y)$，风险部分是 $v(x)-v(y)$，式(4-2)的右边可以化成 $\pi(p)v(x)+[1-\pi(p)]v(y)$。假设 $\pi(p)+\pi(q)=1$，则严格为正或者负的 Prospect 的价值与常态 Prospect 的价值是一致的。但是 $\pi(p)+\pi(q)=1$ 这个条件不是始终成立的，这一点在具体介绍决策权重函数时将会给出解释。

2. 价值函数

价值函数是前景理论用来表示效用的概念，它与传统的标准效用函数的区别在于它不再是财富的函数，而是收益或损失的函数，或者说主观价值的载体是财富的变化而非最终状态，并且这种变化依赖相对于参考点的偏离程度，这一假设是前景理论的核心。卡尼曼和特沃斯基(1979)提出了价值函数与决策权重(decision weight)的模型，以替代期望效用和主观概率模型。价值函数可以表示为

$$V=\sum_{i=1}^{n}w(p_i)v(x_i) \tag{4-3}$$

其中，$v(x)$为决策者主观感受所形成的价值，即价值函数；$\pi(p)$为决策权重，是一种概率评价性的单调增函数。

总的来说，前景理论的价值函数在形式上基本保留了和期望效用函数一样的形式，但其相关因子的含义和度量方式已经与传统期望效用理论有了本质的区别。前景理论一个非常巨大的突破就是用价值函数 v 替换了传统的效用函数，从而将价值的载体落实在财富的变化而非最终状态上。大量的心理学证据表明，人们通常考虑的不是财富的最终状态，而是财富的变化状况。

从总体上看，价值函数有下列四个重要特征。

第一，对于个人来说，任何情况下收益总是比损失要好，而且收益越大(损失越小)，价值越高。因此，价值函数是一个单调递增的曲线。

第二，价值函数是定义在相对于某个参考点(reference point)的收益和损失，而不是一般传统理论所重视的期末财富或消费。也就是说，$v(x)$中的 x 是指相对于参考点的变化，如果没有利得或者损失，则价值为零，$v(0)=0$。在以参考点为原点、以收益为自变量的坐标图上，价值函数是一条通过原点且单调递增的曲线。

第三，根据反射效应，价值函数应该是以原点为中心，向收益和损失两个方向偏离的反射形状，也就是呈 S 形。在面对收益时是凹函数，体现为风险规避，在面对损失时是凸函数，体现出风险偏好的特性。

第四，在价值函数曲线上，亏损区域的斜率大于收益区域的斜率，在图形上表现为：损失状态时的曲线比收益状态时的曲线要陡峭，表明人们对损失所产生的负效用比同等金额的收益产生的正效用大。或者说，虽然决策者在收益域内规避风险，在损失域内寻求风险，但是损失对其造成的心理影响要远远大于收益。

满足这些条件的价值函数呈 S 形并在参考点处最为陡峭，参考点之上为凹，参考点之下为凸，如图 4.6 所示。

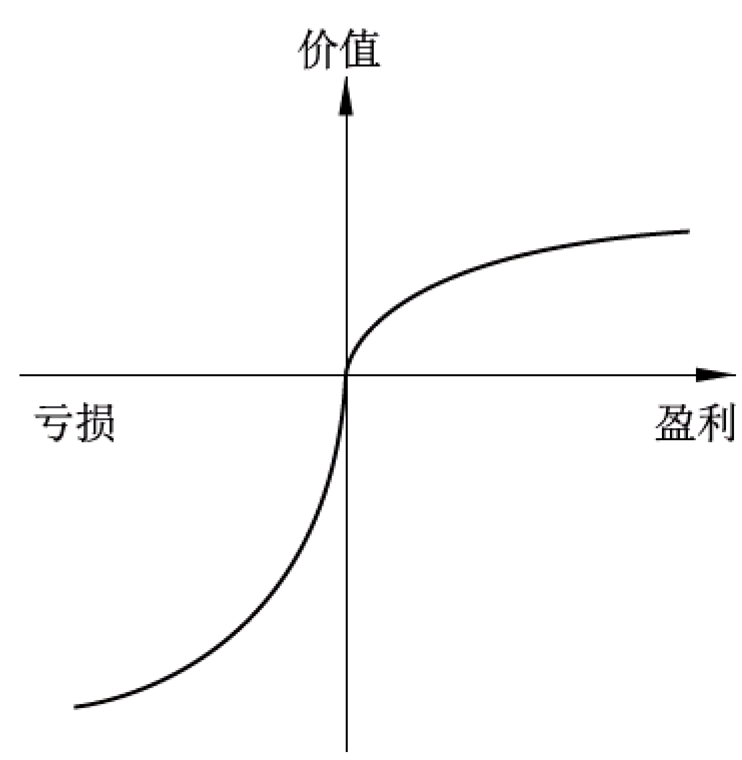


图 4.6　价值函数

卡尼曼和特沃斯基(1979)提出的指数形式的价值函数为

$$v(x)=\begin{cases}x^{\alpha}, & x\geqslant 0\\ -\lambda(-x)^{\beta}, & x<0\end{cases}\tag{4-4}$$

α 和 β 分别表示收益和损失区域价值幂函数的凹凸程度，λ 系数表示损失区域比收益

区域更陡的特征，λ 大于 1 表示损失厌恶。

3. 参考点

前景理论的价值函数和期望效用理论中的效用函数一个重要的不同点是，价值函数存在一个拐点，即存在所谓的参考点。它指人们在评价事物时，总要与一定的参考物相比较，当对比的参考物不同时，即使相同的事物也会得到不同的比较结果，因此，参考点是一种主观评价标准。

参考点的决定通常是以目前的财富水准为基准，但有时也不一定是这样，卡尼曼和特沃斯基认为参考点可能会因为投资人对未来财富预期的不同而有不同的考虑，譬如一个对于损失不甘心的投资人，可能会接受他原来不会接受的彩票。他们发现，在风险条件下，收益机会的价值更多地依赖于可能发生的收益或损失从何种参考点出发，而不是它最终会带来的总财富。卡尼曼和特沃斯基说："我们可以用改变参考点的方法来操纵人们的决策。"

塞勒和 Johson（1990）的一个实验发现，利得会增加个人参加彩票的意愿。他们认为这一结果也可以用参考点来解释。

实验 4-8

参考点实验

A 组学生，假设他们刚刚赢得 30 美元。现在有一个抛硬币的彩票，正面可以获得 9 美元，反面要输掉 9 美元。

实验结果：70％的学生选择接受彩票。

B 组学生，假设刚开始他们没有赢得任何钱，首先就提出了抛硬币的彩票，如果正面则可以获得 39 美元，反面可以获得 21 美元，如果不参加彩票，可以稳获 30 美元。

实验结果：只有 43％的学生愿意参加彩票。

实验结果分析：尽管两组学生面临的选择是一样的，都是正面 39 美元，反面 21 美元，不参加 30 美元，但是刚开始就拥有财富的学生选择参加彩票的更多，而开始没钱的学生放弃彩票得更多。这说明个人做决策时会受到上一次收益的影响。在这里开始状态的 30 美元和 0 美元各自构成了他们的参考点。这也可以用来解释在牛市时，已经从股市获得收益的人，更容易有风险寻求的倾向。

4. 决策权重函数

人们在进行不确定性决策时，要通过概率推理得出适当的结论。概率可以分为客观概率和主观概率两类，客观概率基于对事件的物理特性的分析，如一个硬币向上抛掷，任何一面朝上的概率为 50％。主观概率仅存在于人的头脑中，它是人对事件的客观概率的判断。主观概率往往不是基于对客观情境的分析，而是基于人自己的经验和希望，因此它和客观概率往往不相符合。

期望效用理论认为，一个不确定性期望的价值（效用）可以通过将各个水平的可能结果按照它们出现的概率加权求和得到。假设有一张彩票，当该彩票中奖的概率函数发生变化时，彩票的价值将如何发生变化？根据期望效用理论，彩票的价值是关于中奖概率的线性函数。

但是心理学证据表明，彩票的价值并不是中奖概率的线性函数，从不可能事件到可能事件，或者从可能事件到确定性事件的变化所产生的作用，大于从可能性事件到可能性事件的同等变化而产生的作用，即决策权重存在“类别边际效应”(category boundary effect)。

在前景理论中，每一结果的价值都被乘以一个决策权重，决策权重是决策者根据对预期的选择推断出来的。但决策权重不是客观概率，它不符合概率公理，也不能被解释为个人的主观概率。决策权重函数将期望效用函数中的概率转换成决策权重，决策权重函数 $\pi(p)$ 是概率 p 的一个非线性函数。根据卡尼曼和特沃斯基(1979)的研究，决策权重函数有下列特征。

(1) 决策权重函数 $\pi(p)$ 是客观概率的非线性递增函数，且 $\pi(0)=0$，$\pi(1)=1$。但是 $\pi(p)$ 不是概率，它并不符合概率公理，也不应被解释为个人的主观概率。

除了个人主观认定的事件发生的可能性之外，通常决策权重还会受到与事件相关的其他因素的影响，如个人喜好。人们在做决策时，对自己比较喜欢的结果常常会赋予较大的权重。在购买彩票时，尽管人们知道中奖的可能性很小，但情感的支配使得购买者一厢情愿地认为自己中奖的可能性比较大。

(2) 决策权重函数 $\pi(p)$ 系统性地给小概率事件过高的权重，给大概率事件过低的权重。对于概率 p 很小时，$\pi(p)>p$，这表示个人对于概率很小的事件会过度重视；但是当一般概率或者概率较大时，$\pi(p)<p$，这说明个人在过分注意概率过低的事件的同时，往往忽略了例行发生的事情。

例如，下面的两种彩票，要求被试从中选择其一。

A. 购买彩票，0.001 的概率获得 5 000 美元。

B. 稳获 5 美元。

在 72 个参加实验的人中，72%的人选择了 A。

利用价值函数的方程可以得到：

$$\pi(0.001)v(5\ 000)>v(5)$$

因此，

$$\pi(0.001)>v(5)/v(5\ 000) \tag{1}$$

由价值函数在收益区域的性质(边际收益递减)可知，

$$v(5)>v(5\ 000)\times 0.001 \tag{2}$$

根据式(1)、式(2)，$\pi(0.001)>0.001$

如果将上述实验中的收益改为损失。请被试从 A(0.001，−5 000 美元)和 B(1.0，−5 美元)中选择一种，人们则会选择 B，这表明他们在规避风险。

以上两个实验说明，虽然人们在收益域内通常会规避风险，但是获得收益的概率相当小时，人们就转而寻求风险。虽然人们在损失域内通常会寻求风险，但是当遭到损失的概率相当小时，人们转而规避风险。卡尼曼和特沃斯基将上述现象称为风险态度的“四重性”，该属性表明，人们在面对大概率的收益和损失时，分别会表现为风险规避和风险寻求；而在面对小概率的收益和损失时，人们则分别表现为风险寻求和

风险规避。

生活中有两个熟悉的例子：一是即使中奖的概率很低，也有人热衷于参加高奖额抽奖的现象；二是人们有时会过度支付航空飞行保险费(Eisner 和 Strotz，1961)。

(3) 在低概率区域，决策权重函数具有次可加性(subadditive)，即对任意 $0<r<1$ 时，有 $\pi(rp)>r\pi(p)$。

例如，下面的两种彩票，要求被试从中选择其一。

A. 0.001 的概率获得 6 000 元。

B. 0.002 的概率获得 3 000 元。

实验结果表明 73%的被试选择 A。

根据价值方程，得到 $\pi(0.001)v(6\ 000)>\pi(0.002)v(3\ 000)$，即 $\pi(0.001)/\pi(0.002)>v(3\ 000)/v(6\ 000)$。

根据价值函数的性质，可知 $v(3\ 000)/v(6\ 000)>0.5$。

于是可得：$\pi(0.001)/\pi(0.002)>0.5$，即 $\pi(0.5\times0.002)>0.5\pi(0.002)$。

编者多次以《行为金融学》选课学生作为被试做上述彩票选择实验，实验结果表明绝大多数人选择 A。例如，2018 年 3 月的实验结果为：第一组(72 人)73%的被试选择 A；第二组(53 人)63%的被试选择 A。

(4) 决策权重函数具有次确定性(subcertainty)，即各互补概率事件决策权重之和小于确定性事件的决策权重，即对于所有的 $0<p<1$，有 $\pi(p)+\pi(1-p)<1$，很多实验结果都不同程度印证了次确定性的存在。区间$(0,1]$中 π 的斜率可以被当作对概率变化偏好敏感性的度量。次确定性表明，π 是对 p 的回归，即偏好对概率变化的敏感性通常比期望效用理论要求的低。因此次确定性捕捉到了人们对于不确定性事件风险态度的一个重要组成因素，即与互补性概率事件相关联的决策权重之和小于与确定性事件相关联的决策权重，如 $\pi(0.66)+\pi(0.34)<1$。

例如，下面的两种彩票，要求被试从中选择其一。

A. (2 500，0.33；2 400，0.66；0，0.01)。

B. (2 400)。

实验结果表明 72 个被试中有 82%选择了 B。

根据价值方程式，可得

$$v(2\ 400)>\pi(0.33)v(2\ 500)+\pi(0.66)v(2\ 400)$$

也就是，

$$[1-\pi(0.66)]v(2\ 400)>\pi(0.33)v(2\ 500) \tag{1}$$

如果把上述彩票改为下述彩票，要求被试从中选择其一。

C. (2 500，0.33；0，0.67)。

D. (2 400，0.34；0，0.66)。

实验结果表明 83%的人选择了 C。

由此可知：

$$\pi(0.33)v(2\ 500)>\pi(0.34)v(2\ 400) \tag{2}$$

综合前面式(1)可知：

$$v(2\ 400)>\pi(0.34)v(2\ 400)+\pi(0.66)v(2\ 400)$$

$$1-\pi(0.66)>\pi(0.34)$$

即：$1>\pi(0.66)+\pi(0.34)$。

(5) 决策权重函数具有次比率性(subproportionality)，即对于一个固定的概率比 r，小概率对应的决策权重比率要大于大概率对应的决策权重比率，即有：$\pi(pr)/\pi(p)<\pi(pqr)/\pi(pq)$，对于所有的 $0<p,q,r<1$。因此小概率的结果被过度加权(overweighted)，而大概率的结果则加权不足(underweighted)。如 $\pi(0.1)/\pi(0.2)>\pi(0.4)/\pi(0.8)$。

决策权重函数的非比率性可以解释为什么有些概率性保险(probability insurance)对投资者不具有吸引力。根据传统的期望效用理论，人们在决策时使用的是线性概率，那么，50%概率的下降对应 50%价格的下降，对投资者的效用不会产生影响，概率性保险与普通保险对投资者的效用应该一样。而实验结果说明概率性保险并不受欢迎。根据决策权重函数的非比率性，当概率比一定时，大概率对应的决策权重的比率小于小概率对应的决策权重比率。将损失的概率由 p 降至 $p/2$，不如将损失由 $p/2$ 降至 0 更有价值，这样，人们会认为概率性保险比完全消除风险的正常保险具有更大的风险。如果让人们少购买半年(原来是一年)的保险，保险公司应该给予超过 50%的折扣。广义的概率性保险代表了为减少意外事件发生的可能性而付出确定性成本的措施，在日常生活中，类似于汽车报警器的安装、旧轮胎的更换或戒烟的决定都可以视作概率性保险。人们在日常生活中乐意购买这种广义的概率性保险，正是因为这种保险的价格比根据传统期望效用理论计算出的期望成本更低。因此，由于决策权重函数的非比率性，只有当概率性保险相对于完全保险，其下降的价格高于相应概率下降所计算出的理论价格时才会对消费者具有吸引力。

专栏4.3 概率性保险实验

人们普遍购买保险以防止损失，这是期望效用函数呈凹形的证据。概率性保险代表了为减少意外事件发生的可能性而付出确定性成本的许多种保护性措施。卡尼曼和特沃斯基(1979)设计了一个概率性保险实验，结果表明概率和收益相同的两种预期可能由于表述方式的不同而具有不同效应，这一结果并不符合货币的效用函数在各点都呈凹形的假设。卡尼曼和特沃斯基(1979)在实验中向 95 名斯坦福大学的学生提出以下问题：假设你正考虑是否为某种财产购买保险，以防止火灾或盗窃之类的损害。考察了风险和保费之后，你发现自己在购买保险或让财产处于未保险状态两者之间并无明显的偏好。如果保险公司提供一个新险种，叫作概率性保险，在这个项目中，你付正常保费的一半，损失发生时，你有 50%的机会付另一半保费，保险公司赔偿全部损失；50%的机会你重新得到付出的保费，自己承担全部损失。例如，如果某月的奇数日期发生了一件意外，你付了另外一半保费，损失由保险公司赔偿，但如果事件发生在某月的偶数日期，那么，你已支付的保费被退回，损失由自己承担。在这种情况下，你愿意购买概率性保险吗？

实验结果是 80%的人不愿意购买，只有 20%的人愿意购买。这表明概率性保险不具有吸引力，将损失的概率从 p 减至 $p/2$ 不如将损失的概率由 $p/2$ 降至 0 更有价值。

决策权重函数的上述特征符合心理学原理并得到实验验证，综合以上特征，可以描绘出决策权重函数的近似图像，图 4.7 表示了相对于概率 p 的决策权重函数 $\pi(p)$。决策权重函数是客观概率的非线性函数，单调上升，在低概率段，$\pi(p)>p$，在相对高概率部分，$\pi(p)<p$。即在决策权重函数中人们倾向于高估低概率事件和低估高概率事件，而在中间阶段人们对概率的变化不敏感。

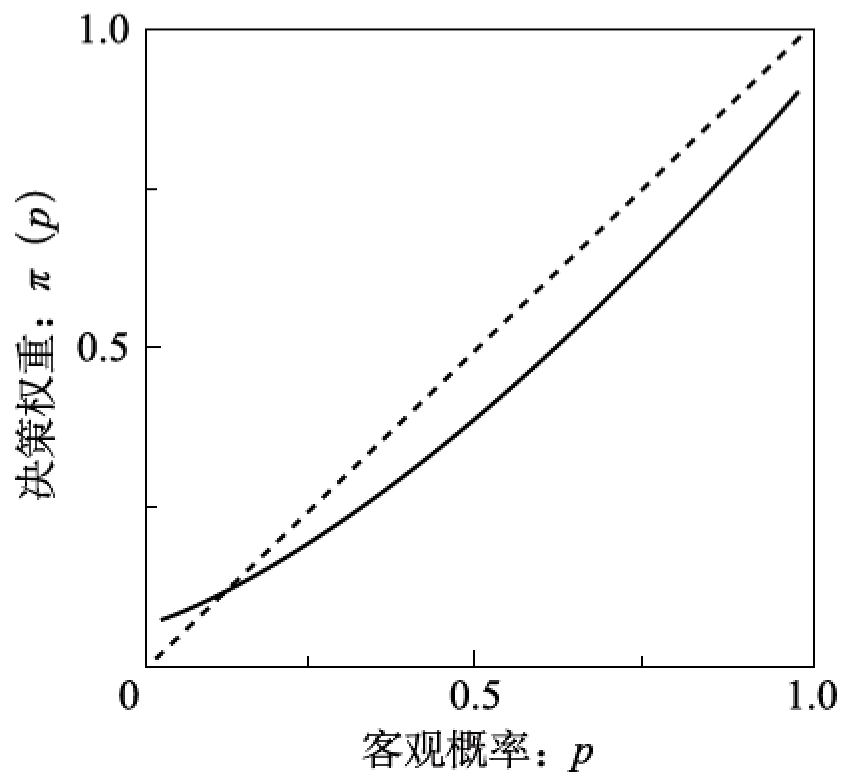


图 4.7 决策权重函数 $\pi(p)$

5. 前景理论的应用

前景理论可以对人们的一些投资行为如“股价溢价之谜”和“期权微笑”作出解释。

1）股价溢价之谜

股票投资的历史平均收益率相对于债券投资高出很多，虽然短期内股票比债券风险大，但从长期来看，实际上是固定收入的长期债券，而不是股票拥有更高的购买力风险，对于从现在开始为退休做 30 年储蓄的人来说，投资于股票市场的风险微乎其微。股价溢价之谜的现象是，相对于债券而言，人们在股票上的投资为何如此之少？

前景理论对此解释如下，投资者在行为模型中是风险回避型的。他们对损失比对收益更敏感，对证券市场价格的频繁波动带有排斥心理，因此在面对这种风险时就要求一个较高的资产溢价。

2）期权微笑

许多关于股票期权定价的实证研究发现一种称为“期权微笑”的现象（Mayhew，1995）。期权微笑，是指当期权处在深度实值（out of the money）和深度虚值（in the money）时，期权价格要高于用 Black-Scholes（1973）期权定价公式计算得到的结果，而在接近真实值的期权（near the money）状态，两者非常接近，形成类似微笑的图形。这意味着，与 Black-Scholes 公式计算的理论价格相比较时，虚值期权和实值期权的定价都相对过高了，接近真实值的期权更接近于准确价格。

现实世界中，期权处于深度实值和深度虚值的概率较低，根据前景理论中的决策权重函数的特点可知，投资者往往高估小概率事件，对小概率事件赋予过高的决策权重。另外，前景理论中期望的价值是由“价值函数”和“决策权重”共同决定的。因此，当投资者对期权深度实值和深度虚值的情况赋予过高的权重时，会导致其对期权的期望价值过高，引

起股票期权价格被高估，出现微笑的现象。

3）开放式基金赎回率

开放式基金通过申购赎回产生的“优胜劣汰”机制对基金管理人形成有效的约束。但我国开放式基金赎回情况与成熟资本市场表现不同，存在基金业绩越好、赎回率越大的“逆淘汰”异象。产生这一现象的原因可能是在基金业绩好的情况下，投资者处于盈利状况，他们更加风险厌恶，倾向于尽快卖出基金份额，变现收益；而在基金业绩差的情况下，投资者处于亏损状态，更加风险偏好，期待业绩在日后能够有所提升，不愿意卖出手中持有的基金份额。这个现象也被称为“处置效应”（deposition effect），即投资者在处置股票时，倾向卖出赚钱的股票、继续持有赔钱的股票，也就是所谓的“出赢保亏”效应。

6. 前景理论与期望效用理论的比较

前景理论与期望效用理论的研究逻辑不同。前景理论对期望效用理论有本质的突破，首先，前景理论中的价值函数与期望效用理论中的效用函数有重大区别；其次，前景理论对风险态度的衡量也区别于期望效用理论；最后，前景理论的突破还表现在对决策权重的改进。具体来说，与期望效用理论相比较，前景理论具有以下特点。

1）决策者关心的并不是财富的绝对水平，而是相对于某一参考水平的变化

卡尼曼和特沃斯基认为，人们在对不确定事件进行判断和估计时，通常会设定一个初值，即参考水平（参考点）。人们在决策时，总是会以自己的视角或参考水平来确定收益和损失，决定各种可能结果的取舍。参考水平常常是决策者当前的财富水平，但也可能是渴望得到的某一水平，即努力获得的财富水平。根据反馈信息，人们对这个初值进行修正，但这种修正往往不完全，似乎锚定于初值。

虽然人们会锚定于参考水平，但与绝对衡量结果相比，人们常常对一个结果与某一参考水平（诸如现状）不同的情况更敏感。也就是说，人们更看重的是偏离参考水平的变化量而不是绝对水平，这与人类的认知心理规律有关。人类的感觉机制是与变化及差别的估计相协调的，而不是与绝对量的估计相协调的。当人们对光、声音、温度等作出反应时，过去和现在的经验确定了一个可接受的参考水平，与此参考水平相对照，目前所处环境中的刺激也就很容易被察觉出来，同样的原理也适用于对健康、声誉和财富等方面的分析。

2）价值函数以财富变化为自变量形如 S

期望效用理论中，函数 u 通常在各处都是平滑和凹的，但价值函数不同。价值函数是以财富的变化为自变量的，其形状是 S 形的。价值函数在参考水平之上的收益区域曲线是凹的，在参考水平之下的损失区域曲线是凸的，这反映了人们在面临损失时不是风险厌恶者，而是风险寻求者。

不论是在收益上还是在损失上，人们的敏感性都呈递减趋势。卡尼曼和特沃斯基（1979）指出，敏感性递减是人类认识的最基本特征。这一特征尤其体现在人们对货币变化的估计过程中，在货币收入不确定的情况下，敏感性递减意味着价值函数对财富变化量的斜率随着财富变化量的增大而变得越来越小。

价值函数并不是处处平滑的，而是在零点（参考水平）有个结。并且，价值函数在损失区域的曲线比收益区域的曲线更陡峭。这种现象与实验心理学的感觉和判断理论是相符

的，反映了人们的偏好有局部损失厌恶的特征。损失厌恶暗含了人们有一种为避免确定损失而愿意冒更大风险的倾向，特别是在面对难以避免的损失时，回避损失能被恐慌所抵消，使人们变得敢于冒险，面对大的损失的风险偏好行为和风险规避的传统假设是不一致的。损失厌恶意味着人们对单位损失比单位利润的评价更高，卡尼曼和特沃斯基估计人们对中等程度的损失的评价大约是同样大小收益的评价的两倍。

3）决策权重是客观概率的函数

期望效用理论是以客观概率为权重，而在前景理论中，决策权重是客观概率的函数。决策权重函数 π 是单调递增的、非线性的，在 0 和 1 处不连续。它系统性地给小概率的权重过大，大概率的权重过小。当概率较小时，一定程度的概率增加，不会较大地改变人们对这些小概率事件赋予的选择权重，此时起决定作用的就是报酬的多寡。反过来，在概率较大时，人们对概率的变化十分敏感，偏好选择中的概率权重变得十分重要。而这些都是金融市场上存在错误定价的原因。

前景理论吸收了不确定条件下选择的一些经验规律，这使它和前述的实验结论相一致，合理解释了不少经济现象，如阿莱悖论、股票溢价之谜以及期权微笑等。后来，卡尼曼和特沃斯基(1992)把期望理论进一步扩展为累积期望理论，克服了它的一些弱点。

以实证研究为基础的累积期望理论允许有多种可能选择的结果，认为应该区分损失和收益，这种区分会改变渐增的概率，而且效用函数也反映了这种区分。卡尼曼和特沃斯基(1979，1992)提出对于收益的效用函数是凹的，而对于损失的则是凸的，而且损失的要比收益的更为陡峭。在累积期望理论的效用函数中，收益是凹幂函数；损失是斜度陡峭的凸幂函数。权重函数是反转的“S”，先凹后凸。累积期望理论利用累积概率而不是个别概率来转换传统效用函数中的概率。累积期望理论认为个人的风险态度有四种不同的类型。当事件出现的概率比较大时，处于收益状态时投资者是风险厌恶的，处于损失状态时投资者是风险寻求的；但当概率很小时，对于收益则形成风险寻求，对损失则是风险厌恶的。总之，前景理论及其扩展，比期望效用理论更精确地描述了个人风险行为。

卡尼曼和特沃斯基(1986)认为，事实上两种理论都是必需的，前景理论和期望效用理论是从两个不同角度考虑决策问题的，二者并不矛盾。期望效用理论描绘了理性行为的特征，而前景理论则描述了有限理性人的实际行为。期望效用理论是公理性的，而前景理论是描述性的，是从经验观测归纳的方式，而不是从一套逻辑上吸引人的公理演绎的方式形成的。期望效用理论为某些简单、透明的决策问题提供了标准；但大多数现实生活中的决策问题是复杂的，需要更加丰富的行为模型。

前景理论及其扩展理论比期望效用理论更能真切地描述风险下的个人决策行为，并且已经在这方面贡献卓著。2002 年诺贝尔基金会网站上指出，前景理论已经成为风险决策研究领域应用性实证工作的基石，它为经济学家们提供了一种新的观察视角，也促进了后来其他理论的发展。不过前景理论也有其不足之处：一是价值函数与决策权重函数都没有给出函数的具体形式，其本身通过举例来说明而带有一定的实验性质；二是价值函数中参考点的确定无一个明确标准。

本章小结

期望效用理论是建立在个体偏好理性的一系列严格的公理化假定基础上的，是现代决策理论的基石，它有4个重要的假设条件：消除性、传递性、占优性和恒定性。但是期望效用理论与真实行为的偏离是广泛的，如难以解释阿莱悖论等现象，没有考虑现实生活中个体效用的模糊性、主观概率的模糊性，不能解释偏好的不一致性、非传递性、不可替代性等现象。随着实验经济学的发展，通过实验论证发现的确定性效应、同结果效应、同比率效应、反射效应、孤立效应和偏好逆转等“悖论”，对期望效用理论形成了巨大冲击。

前景理论认为个人在风险条件下的选择过程可以分为两个阶段：编辑阶段和估值阶段。前景理论的基本模型用价值函数与决策权重函数替代期望效用理论中的期望效用和主观概率模型。价值函数是收益或损失的函数，为S形，在面对收益时是凹函数，在面对损失时是凸函数。表示决策者在处于收益状态时是风险规避的，在处于损失状态时是风险偏好的。决策权重函数是客观概率的一个非线性函数，具有次确定性、次比例性和次可加性等特征。前景理论可以合理解释许多经济现象，如阿莱悖论、股权溢价之谜以及期权微笑等。

思考题

1. 名词解释：确定性效应、同结果效应、同比率效应、反射效应、孤立效应、偏好逆转、参考点。
2. 期望效用理论的公理化假定有哪些？
3. 人们面对风险的主观态度有哪几种？
4. 前景理论中价值函数的主要特征是什么？
5. 前景理论中决策权重函数的主要特征是什么？
6. 前景理论与期望效用理论的主要区别是什么？

第二部分

宏观行为金融学



第5章

并非有效的市场

引导案例：AH股溢价

在我国资本市场中，有些上市公司不仅在境内，也在香港市场发行股票。从理论上来看，同一家公司的股份无论在哪里上市，其价值应该是一样的，价格当然也应该是相同的。然而，相当一部分上市公司在境内和香港市场的价格存在差异。根据2018年8月9日的收盘价格，在总共101家同时在A股和H股上市的公司中，有98家公司的A股价格高于H股价格，仅有中国平安、鞍钢股份、海螺水泥的A股价格略低于H股价格。这就是人们所说的AH股溢价现象。

AH股溢价问题引发了学术界的广泛关注。研究者主要从以下几个角度来解释AH股的差异，包括信息不对称、需求差异、流动性差异以及风险偏好差异。其核心观点是我国A股市场和H股市场是分割的，投资者无法套利，即买入低估的H股，卖出高估的A股，以消除二者的差异。

2014年11月17日，沪港通正式实施，国家允许上海、香港的投资者通过上海证券交易所和香港证券交易所对股票进行双向交易。沪港通的实施理论上来说能够降低两地市场分割的程度，两地股民可通过在香港市场上买入低价的双重上市股，在上海卖出高价的双重上市股进行套利，在获取相同投资标的同时获得无风险套利收益。由此，从理论上可以预测，沪港通实施后AH股溢价应该有所下降。市场的真实情况却与理论预测相去甚远。图5.1统计了2013年至2015年AH股溢价率，结果显示，在2014年11月沪港通实施后，股价差异率并未出现意料中的降低，而是呈现显著提高的趋势。

标准金融学的理论基础之一是有效市场理论，该理论认为即使市场存在非理性投资者的定价偏差，套利能够消除这种偏差，因而股票的定价依然能够有效地反映公司的价值。而我国的AH股溢价不仅长期存在，在实施沪港通、市场分割减弱后，价差反而进一步增强，这表明套利是有障碍的。与AH股溢价现象类似，荷兰皇家和壳牌公司由同一家

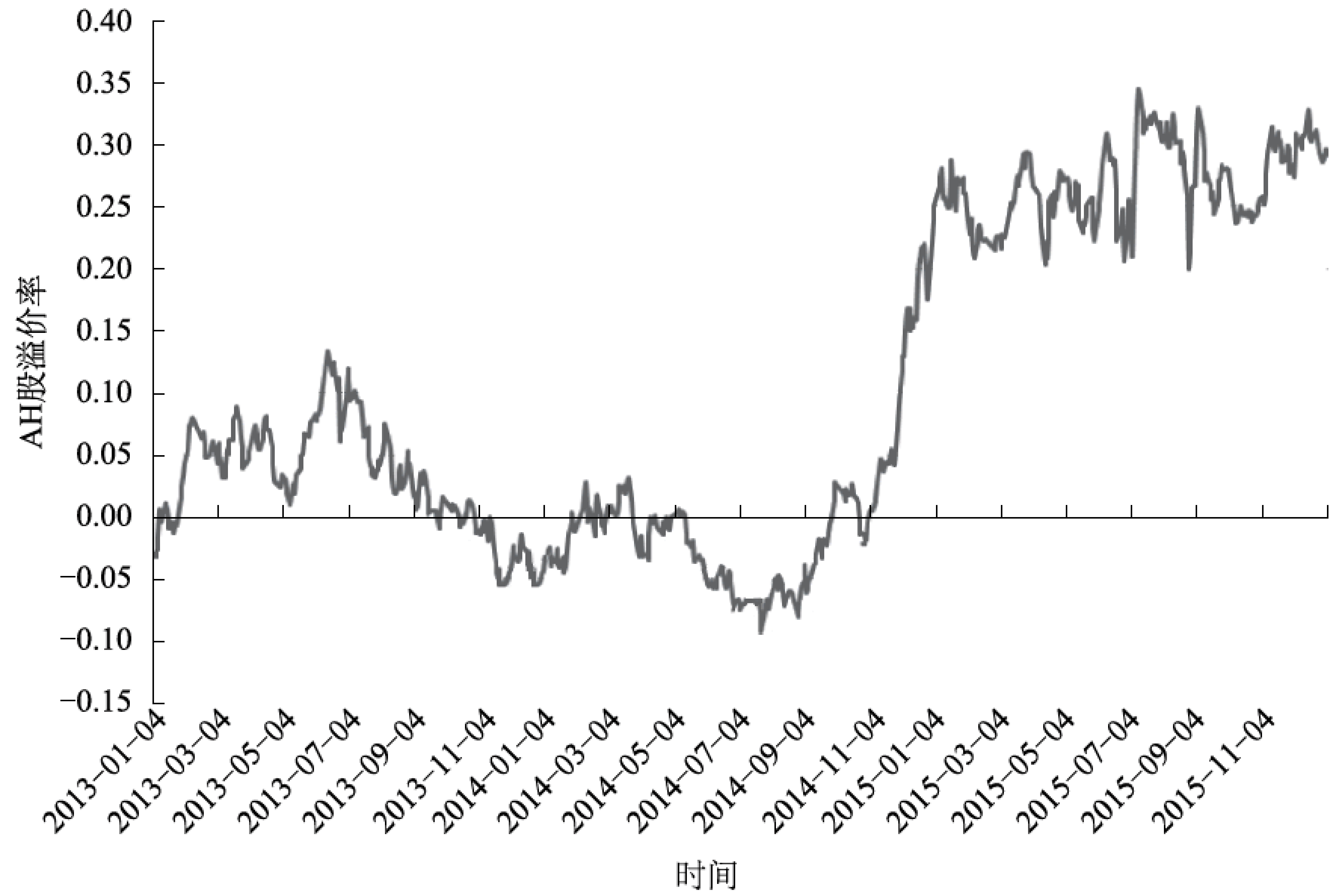


图 5.1　2013 年至 2015 年 AH 股溢价率

注：AH 股溢价率等于(当日 A 股加权平均价格减去 H 股加权平均价格)除以 A 股平均价格。

公司拆分而成，两家公司的股票长期存在价差，这类现象被称为孪生股票定价之谜，也违背了市场有效性的理论。本章将回顾有效市场理论的基本假设与结论，介绍行为金融理论如何对有效市场理论提出挑战，并重点阐述市场套利理论所存在的局限性[①]。

5.1 有效市场理论的基本内容

有效市场理论(EMH：Efficient Markets Hypothesis)认为如果一个市场在决定证券价格时利用了全部信息，并反映在价格上，那么从信息的角度来讲该市场就是有效的。因此，市场的有效性是以理性预期为微观基础的。Fama(1970)的经典论文《有效资本市场：理论和实证研究回顾》不仅对过去有关有效市场的研究作了系统的总结，还提出了研究有效市场的完整理论框架。在此之后，有效市场理论蓬勃发展，其内涵不断加深、外延不断扩大，最终成为现代金融经济学的支柱理论之一。

早期的文献将市场信息完全对称作为有效市场的假设前提，也就是说，投资者能够均等、及时、无成本地获得信息。在进一步的研究中，信息完全对称却被证明不是有效市场的必要条件。理论研究表明，只要有足够数量的理性投资者存在，市场价格就能充分地反映信息，这些理性投资者具备足够的知识和能力去挖掘套利的机会，他们的套利活动使价格迅速地调整到反映完全信息的相应水平。

从本质上来看，有效市场假说讨论的是信息对证券价格的影响问题，包括信息的影响速度和信息集两方面的内容。

---

① 本章部分内容参考史莱佛．并非有效的市场——行为金融学导论[M]．北京：中国人民大学出版社，2003.

1. 速度方面

从市场对信息反应的速度来看，如果证券价格迅速地对所有相关信息作出完全反应，则市场是有效的；如果信息是相当慢地散播到整个市场，而且证券价格可能偏离所有能得到的相关信息所反映的价值，这类市场被称为非有效市场。在非有效市场中，存在反应过度和反应不足两种情况，如图 5.2 所示。标准金融理论更多的是探讨有效市场的情况，对反应过度和反应不足的非有效市场的研究是行为金融学的主要研究对象。

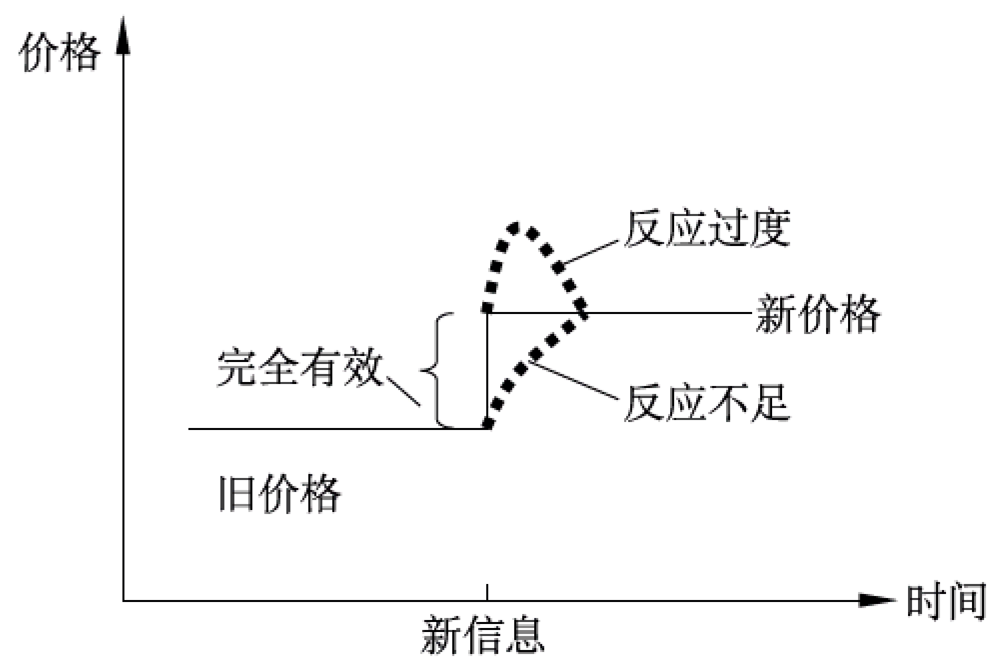


图 5.2　市场对信息的各种反应

专栏 5.1　雄安概念是过度反应吗?

2017 年清明节期间，“雄安新区”横空出世，虽然雄安新区的规划细节尚没有向公众披露，但相关概念股却已被资本市场狂热追逐。雄安新区成立第一个开盘日即 2017 年 4 月 5 日，华夏幸福(600340)、冀东水泥(000401)、金隅股份(601992)、首创股份(600008)、津滨发展(000897)、保变电气(600550)等 20 余只概念股涨停。按照有效市场理论，在消息公布后的 4 月 5 日，股票价格应该充分体现了该利好消息，在之后没有新消息出现的情况下，股票价格应保持稳定。然而，4 月 6 日，雄安概念股板块涨 7.51%，京汉股份、冀东水泥、华夏幸福、金隅股份、河北宣工等 33 只雄安概念股继续涨停。其中的原因可能有两个方面：一是我国股票市场设置了涨跌停制度，单只股票的价格在一个交易日最多只能上涨 10%，雄安概念相关股票在 4 月 5 日可能没有上涨到合理的水平，因而在第二个交易日继续上涨。二是投资者对该利好消息反应过度。从更长的时间段我们可以观察到，初始的狂热跟风之后，投资者开始思考公司与雄安建设的关系有多大、建设周期有多长，与雄安相关的业务什么时候能够体现到公司业绩中。雄安概念相关股票出现了分化，许多股票价格出现了回落，这进一步说明了前期的上涨属于非理性的过度反应。

2. 信息集方面

Roberts(1967)首次按照信息集区分了三种层次的市场有效性：弱势型有效市场、半强势型有效市场以及强势型有效市场。

假设证券市场是竞争性的和有效率的，那么投资者所预期的从某种证券投资中获得的收益将等于所使用资金的机会成本(无风险的利率 r)。如果以 R_t 表示从 t 期到 $t+1$ 期持有某种资产的总收益，有效市场假说认为：

$$E(R_t \mid \Omega_t) = (1 + r_t) \tag{5-1}$$

其中,E 是在 t 时期给定信息集 Ω_t（包括 r_t）条件下的预期。

当 Ω_t 从最小的信息集达到最大的信息集时,证券市场也就从最弱形式的有效市场达到最强形式的有效市场。各种有效市场假说的具体含义是:

弱势有效市场是指信息集 Ω_t 包括了过去的全部信息(历史信息),即过去的信息(如价格和交易信息等)完全包含于当前的价格之中。价格的任何变动都是对新的信息的反应,因此,掌握了过去的信息(过去的价格和交易量信息)并不能预测未来的价格变动。股票市场上大量的技术分析方法主要是基于股票的历史价格、历史成交量,总结历史数据中可能的规律,据此对未来股价变动进行预测。如果一个市场满足弱势有效,投资者借助技术分析方法将不可能获得超额收益,因为所有的历史信息均已包含在股票价格里。

半强势有效市场指当前的证券价格不仅反映了历史价格包含的所有信息,而且反映了所有有关证券的能公开获得的信息。在此假说下,信息对证券价格的影响是即刻实现的。如果一个市场是半强势有效的,任何基于公开信息分析的投资策略都无法获得超额收益。

强势有效市场指在强势型有效市场中,投资者能得到的所有信息(历史价格信息、所有能公开获得的信息和内幕信息)均反映在证券价格上,任何信息都无助于获得超额利润。股票市场的基金通常被认为比普通投资者知识更专业、信息更充分,但在强势有效市场上,专业基金无法获得比普通投资者更高的收益,所有投资者都只能是市场价格的接受者。

三个层次的有效市场信息集之间的关系可以用图 5.3 来表示。

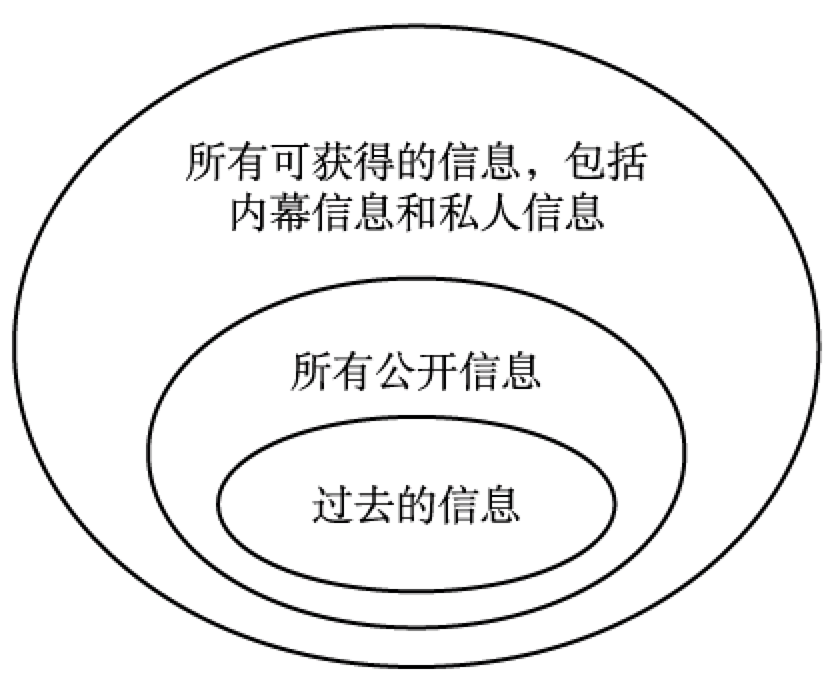


图 5.3　三个层次的有效市场信息集之间的关系

在有效市场理论提出之后,出现了大量对该理论进行检验的实证研究。20 世纪六七十年代大量对金融市场的分析结果支持了有效市场理论。对有效市场理论的检验分析分为两类:第一,价格对信息的快速和正确反应;第二,在无基本价值信息时,价格无反应。在早期的实证研究中,相当多的结论都支持有效市场假设。在有效市场理论获得实证检验支持的同时,反对的意见也接踵而来,人们发现真实证券市场上存在的一些异象是违背有效市场理论假说的,这些异象包括证券市场长期持续的泡沫现象、月份效应、规模效应、封闭式基金折价之谜等,本书第 6 章将专门介绍这些异象。

5.2 行为金融对有效市场理论的挑战

有效市场理论是现代金融理论的基石，在对有效市场理论提出挑战的过程中行为金融理论应运而生。根据Fama(1970)的研究，有效市场假说(EMH)具有以下三个前提条件：第一，证券交易无交易成本；第二，投资者将无偿获得所有信息；第三，所有投资者对信息的解释基本一致。这些条件在现实中都难以满足。Fama(1991)表示有效市场理论在某种意义上是不可检验的。Shleifer(2000)进一步将有效市场假说的理论假设归纳为三个方面：第一，理性投资者假设，投资者被认为是完全理性的，因而可以完全理性地对资产估价；第二，随机交易假设，即使投资者不是完全理性的主体，由于他们的交易是随机发生的，因此交易对价格的影响也相互抵消；第三，有效套利假设，即使投资者非理性且行为趋同，非理性交易行为不能相互抵消，套利者的理性行为仍然可以把非理性交易行为对价格发生的影响冲销。由此可见，有效市场理论的要点是理性(理性预期)导致市场有效性，套利维持市场有效性。行为金融理论从有效市场理论的这三个假设前提出发，利用大量的实验数据、市场数据，提出投资者是有限理性的，投资者的非理性行为是具有相关性的，套利是存在风险的。行为金融理论对有效市场理论假设的挑战如图5.4所示。

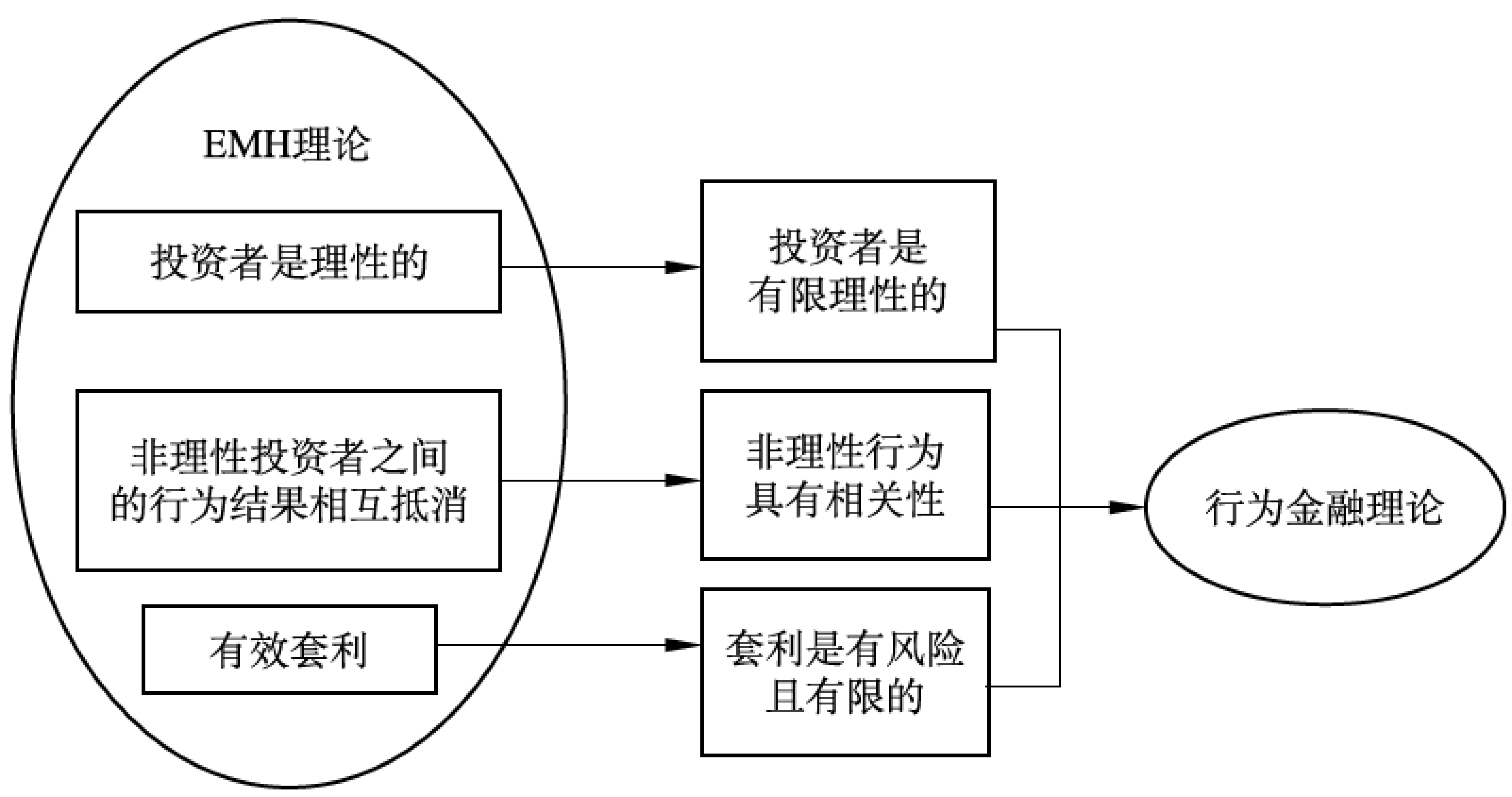


图5.4 行为金融理论对有效市场理论假设的挑战

1. 投资者是理性的吗？

有限理性行为是行为经济学研究的主要内容，在本书的多个章节中都展开过对有限理性行为的讨论。读者可以回顾第2章和第3章，其中的理论详尽分析了人类在决策过程中存在的认知偏差。同样，在面对金融市场的决策问题时，投资者也难以进行完全理性的决策。卡尼曼和Riepe(1998)认为，在许多基本面的假定方面，人们的行为与标准的决策模型是不一致的。这些基本面可以分为三个方面：风险态度、非贝叶斯预期的形成、决策对问题构想和表达方式的敏感性。

1) 风险态度

个人对风险的评价，并不一定遵循von-Neumann-Morgenstern理性概念的假设。在

判断风险时,人们并不看重他们最终获得的财富的绝对水平,而更关注相对于某一参照标准来说他们的得失数量,这一点又会因时因地而变化;同时人们也会尽量避免损失。第4章中介绍的前景理论就是通过很多心理学实验,揭示出不同个体的风险态度存在异质性,同一个体在不同情境下的风险态度也存在差异。

2)非贝叶斯预期的形成

在对不确定性结果进行估值时,个人的行事原则通常会违反贝叶斯原则和其他效率最大化理论。已经有大量的证据表明,投资者的偏好和信念是符合心理学的规律的,而不是标准的经济学模型。例如,人们经常会用短期的历史数据来预测不确定的未来,并试图找出这些过去发生的事情的表征意义有多大。当过分相信这些事件的表征意义时,他们往往忽视这些近期事件的发生仅仅是偶然产生。

3)决策对问题构想和表达方式的敏感性

个体在不确定性条件下的决策受到其作为参考的框架(frame)的影响,问题提出的框架(方式)不同,或个体选择不同的框架对问题进行框定,这些都将导致不同决策结果。个体选择什么样的框架进行框定,是由问题的表达方式和个体的价值观、习惯及人格特质等因素共同决定的。

2. 非理性行为可以相互抵消吗?

心理学的研究成果已经清楚地表明,人们并不是偶然性地偏离理性,而是经常以同样的方式偏离。很多投资者在多数情况下并不是按照自己的投资理念来买卖股票的,他们的买卖行为之间有很大的相关性。投资者之间的交易也并非随机进行,而是在大致相同的时间,大家都试着去买或者卖相同的股票。Shiller(1984)认为,由于受传闻的影响,或者大家都去模仿周围人的行为,噪声交易者的行为就有一定的社会性,大家就会犯同样的错误。投资者心态理论讨论的就是大量投资者犯同样的错误且他们的错误又有相关性的现象。

个人投资者不是唯一的非理性投资者。在发达的金融市场中,大量的资金由共同基金、养老基金的专业经理来管理。他们也会产生个人投资者可能的误差。同时,他们又是管理他人资金的代理人,这种委托—代理关系实际上会带来决策中更大的偏差。在现实市场中,基金经理人的薪酬很可能是由相对业绩决定的,也就是由他个人业绩在市场中的排名决定的,因此经理人倾向于选择与其他经理人一样的资产,以避免在绩效排名中落后。例如,在年末时,他们会不约而同买入最近业绩好的股票,抛掉业绩差的股票使基金的业绩看上去好一些,但是这些决策实际上偏离了资产价值的最大化原则,只是一种"粉饰"(window dressing)。这时基金的偏差行为实际上也是具有系统性和群体性的。

本书的第10章将对证券市场群体性的非理性行为——证券市场羊群行为(herding)进行讨论。

3. 套利是有效的吗?

套利是金融学的基本概念之一,是指在两个不同的市场中,以有利的价格同时买进和卖出同种或者本质相同的证券的行为。从理论上看,套利行为既不需要付出资金成本,也不用承担风险。举一个简单的例子,假设某一天有人发现在A国市场1美元可兑6.7元人民币,在B国市场兑1美元需要6.9元人民币。他可以在A国用人民币买入1美元,在

B 国卖出 1 美元，获得 0.2 元人民币的套利收益。

有效市场理论的最后一道防线是基于套期保值的有效市场。如果套期保值能够抵消非理性投资者的偏差，市场依然有效。正如上面的例子，当理性的投资者发现了 A 国和 B 国在汇率上的差异，会大量在 A 国市场买入美元，在 B 国卖出美元，这种套利行为会最终导致 A 国美元需求上升，汇率上升；同时，B 国美元需求下降，汇率下降，最终 A、B 国的汇率趋于一致，调整到整个市场供需决定的均衡水平。理论上来看，在证券市场也可以进行这样的套利操作。例如，假设由于受到非理性投资者过度乐观的影响，A 股票的价格被高估了，理性的投资者可以识别出 A 股票价格过高，他们在市场上卖空 A 股票，同时买入与 A 股票类似，但价格更低的 B 股票。通过这样的操作，A 股票价格趋于下降，B 股票价格则趋于上升，最终二者都达到合理的水平。这就是无风险套利维持市场有效性的过程。

从理论上看，套利能够消除市场的定价偏差，但真实市场的套利行为是有限的、有风险的。例如，人们会发现在两个市场买卖外汇都有交易费用，套利是有成本的，而且有可能资本不能在两个分割的市场中自由流动。下一节将对证券市场套利的有限性展开讨论。

专栏 5.2 2013 年诺贝尔经济学奖——有效市场 VS 非有效市场

2013 年度诺贝尔经济学奖在当地时间 10 月 14 日揭晓，美国经济学家尤金·法马、芝加哥大学教授拉尔斯·皮特·汉森以及美国经济学家罗伯特·J. 席勒获奖。瑞典皇家科学院说，三名经济学家为“现有对资产价值的认知奠定了基础”。事实上，三名经济学家对于资产定价的观点不尽相同，特别是法马和席勒的观点是完全针锋相对的。

Fama 教授是有效市场理论的创始人，他在 1970 年提出了有效市场假说，1992 年还与 French 共同提出“Fama-French”三因子模型，对资本资产定价模型进行改进。法马被誉为“现代金融之父”，是传统金融理论的代表性人物。

席勒则致力于用市场数据证明市场是非有效的，他发现美国股票的价格长期偏离理性预期的水平，代表作《非理性繁荣》则系统地阐述了非理性投资者如何导致资本市场泡沫的膨胀和破灭，他是行为金融理论的领军人物。

传统金融理论和行为金融学的代表人物同时被授予诺贝尔经济学奖。评选委员会在当天发表的声明中说，获奖者的研究成果奠定了人们目前对资产价格理解的基础，资产价格一方面依赖波动风险和风险态度；另一方面也与行为偏差和市场摩擦相关。这说明资本定价如此复杂，股票价格既需要用理性的观点去解释，也需要用非理性的视角去看待。两种理论共同推动了现代金融学的演进。

5.3 套利的有限性

无风险套利只是一种完美的假说，有非常严格的前提条件，主要包括以下几个方面：第一，能够找到同质的、可替代的资产；第二，套利是无风险、无成本的；第三，市场及其制度结构是完美的，包括信息完全对称、没有市场分割、有允许卖空的制度安排等。真实市

场与前提假设的差异，将使套利存在局限性，真实市场与有效市场也将相去甚远。

1. 完全可替代的资产

套期保值的有效性取决于是否存在近似的替代资产。为了回避风险，套利者在卖出或者卖空价格被高估的证券的同时，必须能买进同样或者相似且价格没有被高估的替代证券。在上文提到的汇率的例子中，在不同市场上的美元是完全同质的，即可以完全替代，如果不考虑交易成本和资本流动限制，外汇市场比较容易实现无风险套利。一些金融衍生资产（如期货、期权等）也有近似的、可替代的资产，如期货市场和现货市场的沪深300指数产品是比较近似的资产，但是许多资产并没有替代资产。一旦由于某种原因证券出现了定价偏差，套利者无法进行无风险的对冲交易。即使某个套利者发现总体股价已经被高估，他也无法卖空并买进替代的证券组合，因为找不到这样的组合。套利者只能简单卖出或者减持风险已高的股票，以期获得较高的收益，但这种套利已非无风险套利。此时套期保值不能保持价格水平，在资产价格发生偏差时，对套期保值者而言不存在无风险的套利策略，他们将价格保持在基本价值附近的能力也受到了限制。

相对于整个市场指数产品而言，单只股票的替代品也许更容易找到，如行业类似、规模类似的股票，但是与单只股票的基本价值相关的风险对套利还是形成了很大的障碍。当一个套利者依据相对价格的变化购进或者卖出股票后，他要承担这种单只股票相关的风险。例如，在我国牛奶行业的上市公司中有伊利、光明、蒙牛等知名公司。假设一个投资者认为相对于伊利的股票价格，光明估值较高，他进行了卖出光明、买入伊利的双向操作。通过这样的操作能够规避整个市场的系统性风险以及与牛奶行业相关的行业风险，即使市场指数下降或者牛奶行业出现负面新闻整体下滑，该投资者的对冲交易也使其不会遭遇任何损失。然而，如果光明业绩意外上升或者伊利业绩意外下降，且均与市场和行业无关，则该投资者将遭遇损失。例如，在2016年的奥运会中，由光明乳业赞助的中国女排在决赛中以3∶1战胜塞尔维亚队，历史上第三次捧起奥运会冠军奖杯。奥运期间尤其是女排决赛至最终夺冠的期间，光明莫斯利安酸奶的日均销量较平时最多增长41.7%。女排夺冠当日，光明乳业的股票几近涨停。从这个例子中可以看到，要想找到完全相同的股票进行套利操作是非常困难的。

2. 噪声交易风险

根据标准金融理论，资产的估值等于其未来收益流的贴现值，完全理性的投资者能够根据市场信息对资产进行正确的估值。噪声交易者是指对风险资产未来收益分布形成错误理念的投资人。套利理论认为理性投资者可以利用噪声交易者的错误定价获得套利收益，他们执行的投资策略是将证券价格推回到与证券基本价值相符的水平上，但这并不意味着他们能够完全实现这个策略。

理性投资者发现了资产价格存在的偏差，但他们很难判断这个偏差会进一步扩大还是会反转。噪声交易者也许会进一步走向极端，使价格进一步偏离合理的水平。这种由噪声交易者心态变化导致对正常状态偏离更远的风险，被称为噪声交易者风险（noise trader risk）。任何在短期进行套利的投资者都会面对这种风险。简单来说，假设目前噪声交易者过度乐观，推高了股价，套利者在价格偏离5%的时候就开始卖空，然而他可能会

发现噪声交易者继续把股价推进到偏离10%的水平，套利者判断错误，开始承担损失；噪声交易者占据上风，股价推高到偏离15%，套利者认为过度高估，继续选择卖空；但很可能噪声交易者强烈看好后市，价格进一步扭曲，套利者又亏损。有鉴于此，在面对噪声交易者时，套利者不得不控制自己的投资欲，为应付将来面临的风险留足头寸。

Shleifer(2003)曾经以美国股市为例对套利的局限性和风险性进行分析。在1998年年底，美国大型公司股票的市场价格相对于按多数指标计算的盈利水平都创了历史新高。如构成S&P500指数的成分股的市场价格与对应的公司盈利水平的比率大约为32，而第二次世界大战后的平均值为15。一些著名的经济学家，如美联储主席格林斯潘，早在1996年就提出警告要注意这类股票的股价高估问题。但他们的警告也受到一些机构的反驳，如高盛公司的Cohen就认为，美国的大公司处在低风险迅速扩张的新时代，所以保持高的市场价格是合理的。

套利者如果按他们所说的那样去做，结果会如何呢？假设一个套利者在1998年年初卖出S&P500股指空头，当时该指数的市盈率倍数已经在24的水平之上，那么到年底他将损失28.6%；如果更早一点，他在1997年年初就按照专家的意见进行卖空，他在当年将损失33.4%；如果他采用一个对冲交易，在1998年年初卖出S&P500股指的同时买进小公司的罗素(Russell)2 000股指作为对冲，后者按照历史标准未创历史新高，那么在1998年年底他将损失30.8%。S&P500股指并没有很好的替代品，选择不合适的替代品进行对冲，可能会使结果更差。

1996年，美国经济出现了前所未有的一系列现象：创纪录的长期繁荣、高增长，甚至出现低通胀伴随着低失业率。人们似乎感到了一个不同于过去的经济形态的出现。在投资者乐观情绪的推动下，股票市场持续上涨，直至2000年网络股泡沫破灭，才使整个市场的价格发生反转。也就是说，理性的投资者如果采用套利策略，要到2000年才能成功。这个例子说明利用股指套利的风险非常大。

3. 套利的成本和制度限制

套利并不是无成本的，套利交易买入和卖出证券都需要交易费用。特别是卖出交易，套利者并不是卖出自己已有的资产，而是需要先从证券机构借入该资产，再在市场上卖出，这种交易行为被称为卖空(short sell)。借入证券是需要成本的，这成为套利成本的一部分。如果将这些成本考虑进去，获取套利收益并不是一件容易的事情。

在许多市场中，卖空是被禁止的。从理论上来看，卖空是维持市场有效性的一种必要手段。一些理论研究认为，如果限制卖空，对市场未来持悲观态度的投资者只能选择离开，导致资产价格不能有效地吸收和反映负面信息。然而，在金融市场实践中，卖空被认为会加大市场的波动，尤其是在市场处于剧烈下跌的过程中，卖空机制很可能会加速股指的崩盘。监管当局往往在股市下跌时，对卖空加以限制。

我国于2010年才推出了融资融券试点政策，允许部分试点机构开展融资交易和融券交易业务，融资融券的标的物也仅限于整个A股市场的部分股票。融资交易就是投资者以资金或证券作为质押，向证券公司借入资金用于证券买入，并在约定的期限内偿还借款本金和利息；投资者向证券公司融资买进证券称为“买多”。融券交易是投资者以资金或

证券作为质押，向证券公司借入证券卖出，在约定的期限内，买入相同数量和品种的证券归还券商并支付相应的融券费用；投资者向证券公司融券卖出称为“卖空”。

除了卖空限制之外，对资本流动的管制、约束以及市场的分割都可能造成跨市场的套利难以实现。例如，在本章开头提到的案例中，同一家公司在 A 股和 H 股的价格存在差异，其中一个原因可能是与市场分割有关，两个市场的投资者不同，资金不能完全自由流动，通过套利来消除价差存在一定的限制。

专栏5.3 西班牙禁止卖空股票

从 2011 年到 2012 年，西班牙 IBEX35 股指持续下跌，成为欧洲主要股指中表现最差的一个。2011 年 8 月，西班牙股票市场监管机构(CNMV)开始禁止市场做空西班牙银行股；2012 年 7 月，将“禁空”范围扩大至所有股票。上述措施原定于 2012 年 10 月 19 日结束，但西班牙监管机构将其延长至 10 月底，并于 11 月 1 日宣布股票卖空禁令再延长 3 个月，直至 2013 年 1 月 31 日。CNMV 表示，延长“禁空令”是考虑到目前西班牙银行业面临“异常情况”，取消该措施将“增加不确定性”。

5.4 有效市场理论的实证检验

自从 Fama 1970 年提出有效市场理论以来，利用市场数据对有效市场理论进行检验一直是金融学研究的热点问题之一。研究者正是在这样的研究中发现了大量与有效市场理论不符的现象，并将这些现象称为“异象”。对这些异象的解释，推动了传统金融理论和行为金融学理论的发展。本节将介绍检验有效市场理论的几种思路。

专栏5.4 足球俱乐部股价对足球队比赛输赢结果的反应

Palomino 等(2009)利用 16 家在伦敦股票交易所上市的足球俱乐部数据，考察了 1999—2002 年俱乐部所属球队比赛的胜负是否影响股票价格。样本期间，16 家俱乐部参与 916 次比赛，405 场胜，235 场平，278 场输。检验发现，球赛后 3 天，获胜球队获得 88 个点的平均累计超额收益率；约 60%的累计超额收益率来自第一天。

输球队产生－101 点的累计超额收益率，赛后第一天仅占 28%。这说明利好消息的反应速度超过了利空消息。

1. 对有效市场三种形式的检验

1）对弱势型有效市场假说的检验

弱势型有效市场条件下，任何历史信息均无法预测股价的走势。据此，对弱势型有效市场的检测一是要看连续的价格变化是否独立；二是看根据历史信息能否获得超额利润。检验的方法可以包括模拟检测、系列数据的相关检测、趋势检测等。

2）对半强势型有效市场假说的检验

半强势型有效市场下，股票价格包含了所有的公开信息，包括历史信息和有关未来的信息。检验的思路在于判断所有公开信息是否迅速、充分地反映在股票价格上，以及使用

这类信息能否获得超额投资利润。例如，上市公司公布年报、红利政策、新产品开发计划、并购计划……市场是否能迅速反应？本书第8章会详细介绍一些相关研究，探讨股票价格对各种公开信息是反应不足、反应过度，还是有效反应。

3）对强势型有效市场假说的检测

根据强势型有效市场的定义，所有信息，包括公开的和未公开的能够全部反映在证券价格上，拥有内部信息的人也不能获得超额利润。对强势型有效市场的一个检验方法就是以共同基金为研究对象，考察它们能否获得超额收益。与普通投资者相比，共同基金不论是在投资的专业技能上，还是在信息上都具有优势，有专业团队收集信息、分析信息，有可能获得一些普通投资者未知的内幕信息。如果基金尚不能根据其信息优势得到超额收益，就在一定程度上表明市场是强势型有效。在早期，Friend，Brown，Herman 和 Vickers (1962)研究共同基金的业绩，结论是共同基金的业绩总体上不比市场的平均收益高或低。Sharpe(1966)，Jesen(1968)，Firth(1978)，以及 Friend，Blume 和 Crokett(1970)相继对基金作出相关研究，结果都支持了强势型有效市场假说。

2. 无消息是否反应

根据有效市场理论，价格只对信息作出反应。检验有效市场理论的第二个思路就是看如果没有实质性的信息，股票价格是否会发生变化。研究者常用的事件包括以下两个方面：①股票更名；②股票被纳入某个指数。在这两种情况下，与上市公司有关的基本信息其实并未发生改变，理论上股价也不应该发生变化。然而，相当多的实证表明，这两类消息公布之后，股票会产生异常收益。这两种情况分别被称为更名效应、指数效应。

在2000年的网络浪潮中，美国相当多的上市公司更名为互联网公司，即使主营业务不是互联网行业，甚至目前从事的业务和互联网没有任何关系。Copper 等(2001)研究了这种更名对股票价格的影响。他们发现公司改名为与互联网相关对股价有显著的正面影响。更名消息发布10天内，产生非正常收益累计达74%。此外，无论公司与互联网紧密度如何，更名消息效应相似。这个现象违背了有效市场理论，主要原因是当时投资者存在对互联网的过度乐观。

专栏5.5 “多伦股份”更名为“匹凸匹”

2015年互联网金融、P2P成为我国金融市场的热点。上市公司多伦股份更名为“匹凸匹金融信息服务股份有限公司”，简称“匹凸匹”，与“P2P”概念谐音。在互联网金融风口上，更改的名字直接为公司股票带来了6个涨停板，涨幅达77%。然而，这个自称更改主业为互联网金融的公司却从未从事过任何此类业务。不仅如此，更名后公司绩效持续下滑，连续两年业绩净利润为负，成为ST匹凸，*ST匹凸。

2017年*ST匹凸发布公告称，拟将公司名称由“匹凸匹”变更为“上海岩石企业发展股份有限公司”，公司简称变更为“岩石股份”，且经营范围剔除互联网金融信息服务业务。宣布更名后的第二个交易日，*ST匹凸放量涨停，最终股价大涨4.96%。

更有意思的是，这家公司的名字已经经历了11次变更：福建豪盛→ST豪盛→福建豪盛→利嘉股份→G利嘉→G多伦→多伦股份→匹凸匹→ST匹凸→*ST匹凸→岩石股份。

3. 实验室实验与准自然实验的方法

在前文介绍套利局限性时已经指出一些制度约束会阻碍套利的实现。一个检验市场是否有效的方法是看如果放松套利约束,市场效率能否提高。如果套利约束的放松能带来市场效率的提高,在一定程度上证明基于套利机制的有效市场理论是可以成立的。

实验室实验方法是在可控制、可重复的实验条件下,通过随机分组、设计制度变量,检验某个制度安排对实验结果的影响。借助实验方法,可以构建虚拟实验室股票交易市场,评估某项制度的效果。Haruvy 和 Noussair(2006)在实验室股票市场中检验了卖空对市场定价效率的影响。他们发现,在没有卖空机制的市场,股票价格高于其基础价值,出现了泡沫现象。在有卖空机制的市场中,价格显著降低了,但价格依然没有收敛到基础价值的水平。Haruvy 和 Noussair(2006)的研究表明套利不能完全消除市场的定价偏差。

准自然实验方法是将真实验的思路用于解决实际问题的一种研究方法,它不能完全控制研究的条件,在某些方面降低了控制水平。虽然如此该方法却是在接近现实的条件下,尽可能地运用真实验设计的原则和要求,最大限度地控制因素,进行数据处理和分析,因此准自然实验研究的实验结果较容易与现实情况联系起来。准自然实验方法是检验制度设计效果的一种有效的手段。例如,一个市场的相关制度安排发生变化就可以天然地成为准自然实验过程。

我国沪深两市于 2010 年 3 月 31 日推出的融资融券业务就为研究卖空机制对资产定价效率的影响提供了难得的自然实验环境。李志生等(2015)将我国股票市场套利约束放松作为一个制度变量,用准自然实验的方法,检验了套利约束对市场有效性的影响。他们利用 2009 年 4 月至 2013 年 12 月中国 A 股市场的数据,通过比较融资融券标的股票和非融资融券标的股票,以及股票加入和剔出融资融券标的前后的定价效率,发现融资融券交易的推出有效改善了中国股票市场的价格发现机制,融资融券标的股票的定价效率得到了显著提高。此外,融券卖空交易量越大,股票的定价效率越高。这一结果说明,套利约束能在一定程度上提高市场效率。尽管如此,基于种种限制条件,现实市场与完全有效市场还相去甚远。

虽然已有的一些研究支持套利能够对提高市场效率起到一定作用,行为金融学大师史莱佛(2003)提出了一个明确的观点,那就是即使在接近教科书模型的完美环境下,套利的影响也是非常有限的;在更为复杂的真实市场环境中,其作用会更小。

本章小结

有效市场理论认为市场价格能够迅速反映市场所有的信息。根据价格反映信息集的不同,有效市场可以分为弱势、半强势、强势三种类型。有效市场理论的成立依赖于三个假设:第一,市场投资者是完全理性的;第二,即使存在非理性的投资者,他们的非理性行为是随机的,而且能相互抵消;第三,即使非理性行为不能相互抵消,市场上理性的投资者也可以通过套利活动消除市场定价偏差。套利被认为是维护市场有效性的最后防线。

行为金融理论对有效市场理论提出了挑战,基本观点包括个体是非理性的、个体之间

的非理性行为是相关的、市场套利是有限制的。本章重点阐述了套利的局限性，包括套利的风险、成本和制度约束等。在实证研究中，研究者已经用实证、实验等方法对有效市场理论进行检验。

思考题

1. 请分析我国股票市场是否满足有效市场的前提条件。
2. 举例说明套利的局限性。
3. 请设计一个实验室实验用于检验套利操作对市场有效性的影响。

第6章

证券市场中的异象

引导案例：天气与股票收益

Hirshleifer 和 Shumway(2003)研究了1982—1997年26个国家的股票交易所所在城市的天气与市场指数日收益率之间的关系。他们用当地时间每天早上6点到下午4点的云层覆盖率作为天气的代理变量，依据天空被云层覆盖区域的比例，用数字0~8代表天空云层覆盖率，数字0表示天气晴朗无云(clear)，数字8表示云层完全覆盖(overcast)。研究发现，日照与股票收益有显著相关性，而下雨和下雪则对市场指数日收益没有显著影响。从所有城市的数据分析看，日照越多，即天空云层覆盖率越小，股票市场指数日收益率越高，(天空云层覆盖率对市场指数日收益的回归系数为−0.011，在10%的显著性水平下显著)如图6.1所示。

传统的有效市场理论下，股票市场是理性的，其反映的仅是有关资产定价的信息，而股票市场中的投资者也是理性的，总是能够最大化其预期效用。天气也许能够影响农业和与天气相关的公司，但在现代经济中，农业扮演的角色已没有以往那么重要。那么，天气是晴天还是阴天并不会影响交易所市场指数的理性价格。

然而大量的实证研究表明，天气的确对股市收益产生了显著影响。那么，天气是通过什么来影响股票收益的呢？心理学研究和生活经验都表明晴朗的天气与人的积极情绪有关。晴朗的天气影响了人们的情绪，而情绪会影响人的判断和行为，人们在积极的情绪下对未来的预期往往比在消极的情绪下更为乐观。

在证券市场中，类似天气带来股市异常收益的现象并不少见。本章总结了证券市场中的典型异象，主要有股权溢价之谜、封闭式基金折价之谜、日历效应、规模效应、账面市值比效应、投资者关注度、本地偏好和无信息反应效应。根据有效市场假说，如果价格可以及时且充分地体现所有的信息，那么无论投资者采取何种策略，都无法获得超额收益。而证券市场中这些异象的存在极大地挑战了有效市场假说，传统的金融理论也无法合理

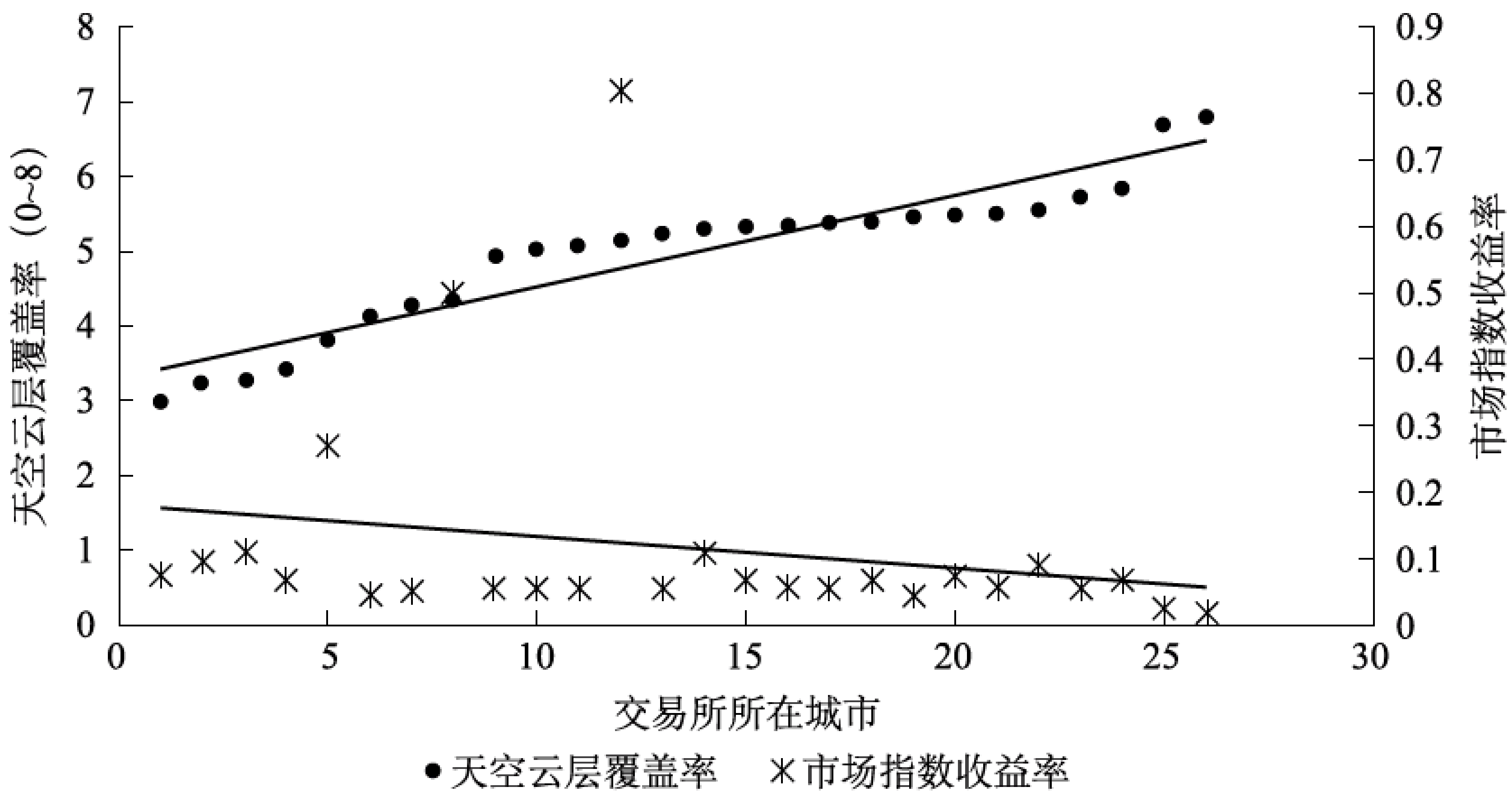


图 6.1　天空云层覆盖率与股票收益率散点图

资料来源：HIRSHLEIFER D A,SHUMWAY,T. *Good day sunshine*:*stock returns and the weather*[J]. *Journal of finance*,2003,58(3):1009-1032.

注：横坐标 1～26 代表的是 26 个国家和地区的股票交易所所在城市按其平均天空云层覆盖率(sky cover)从小到大进行排序，分别为约翰内斯堡、雅典、圣地亚哥、马德里、伊斯坦布尔、悉尼、米兰、布宜诺斯艾利斯、纽约、维也纳、布鲁塞尔、里约热内卢、巴黎、马尼拉、苏黎世、哥本哈根、阿姆斯特丹、奥斯陆、赫尔辛基、斯德哥尔摩、曼谷、中国台北、伦敦、都柏林、新加坡和吉隆坡。

解释。行为金融学将心理学与金融学相结合，对这些异象作出合理的解释。

6.1　股权溢价之谜

股权溢价是指风险证券的收益高出无风险债券收益的部分。传统金融理论认为，溢价是对风险的补偿，持有股票的风险大于持有债券的风险，投资者必然要求更高的回报。然而现实中股权溢价的部分已经超出了标准新古典主义范式下的合理解释范围，所以这远远超出资本资产定价模型(CAPM)数值的溢价部分从何而来，便形成了股权溢价之谜。

Mehra 和 Prescott(1985)在其文章中提出了股权溢价之谜，此后股权溢价之谜成为资产定价领域经久不衰的议题。Mehra 和 Prescott 研究了 1889—1978 年这 90 年间的标准普尔指数，发现其年平均实际收益率为 6.98%，而同期的短期债券的年平均实际收益率仅为 0.8%，即年平均溢价为 6.18%。而依据 Breeden(1979)基于消费的 CAPM 对这一现象难以进行解释，因为历史数据表明同一时期的美国消费增长波动性很小。如果根据模型推算，投资者风险厌恶系数将达到 30～40，远远高于合理水平(10 左右)。

为何美国的证券市场会有如此高的股权溢价？如图 6.2 所示，在 1990—2005 年，美国股权溢价一直维持在较高水平，最高的时候甚至接近 60%。直观的解释是股票比无风险证券承担更高的风险，美国股票市场每年收益标准差为 20%左右，而债券市场仅为 4%。股票投资者因其承受额外风险而要求更高的风险补偿，即通常所说的高风险、高收益。然而值得思考的是，尽管股票的风险远远高于无风险证券，但 1926 年以来高达 8.63%的股权溢价，根本无法用传统的资产定价模型估计的数值解释，股票超额收益之高令人吃惊。

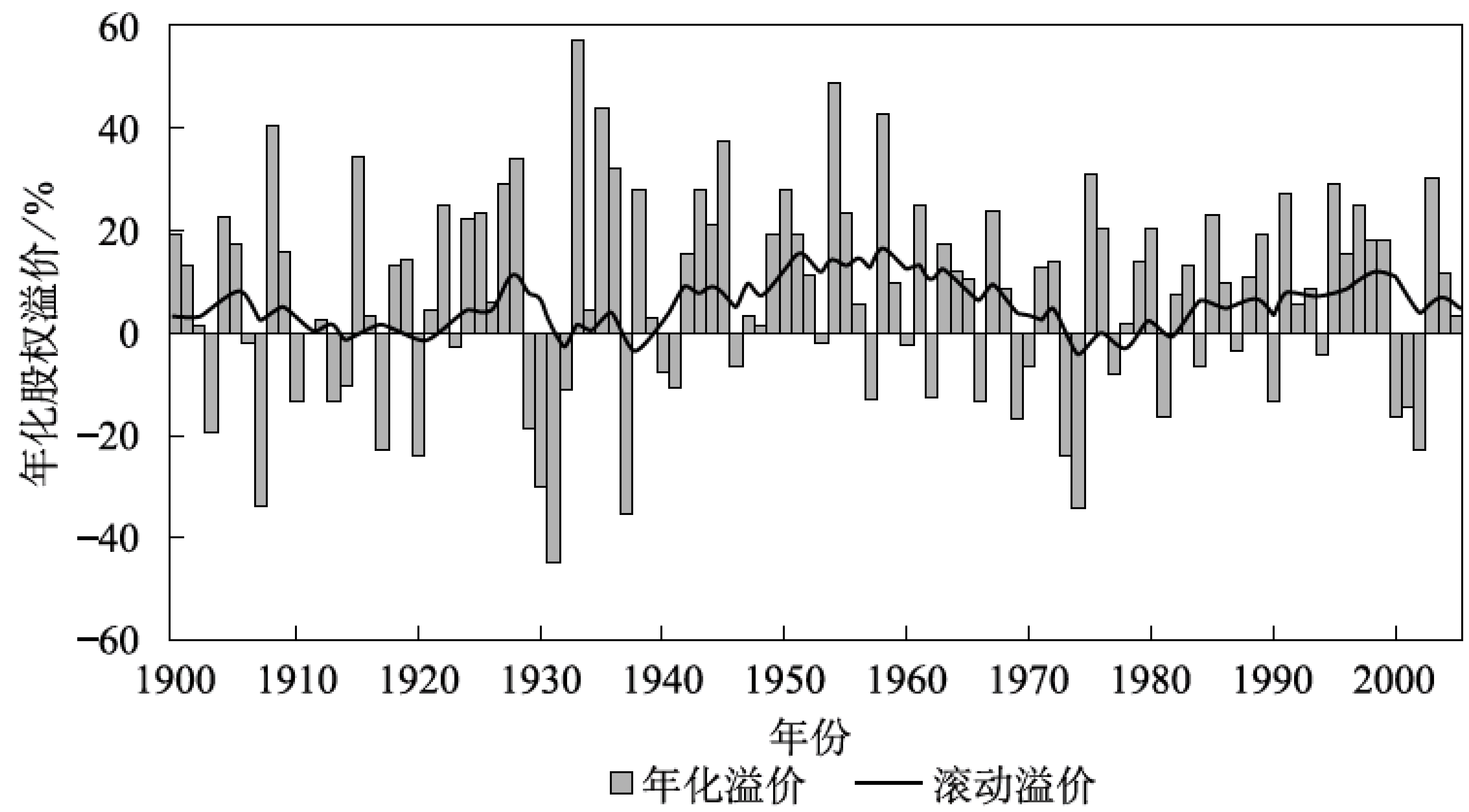


图 6.2　美国 1900—2005 年折合成百分比的年化股权溢价数据

资料来源：DIMSON E，MARSH P，STAUNTON M. *The worldwide equity premium：a smaller puzzle*[M]. Chapter 11 of R Mehra(Ed)，Hand book of the Equity Risk Premiwm Elsevier，2008：467-514.

从风险—收益角度出发显然无法解释股权溢价之谜，随着研究的深入，有学者从行为金融角度对股权溢价进行了合理的解释。Benartzi 和 Thaler(1995)认为人们在选择投资组合时，会计算每一种资产潜在的收益和损失，然后选择期望效用最高的一个。同时投资者存在短视性损失厌恶，会格外关注资产组合的安全性并且频繁对投资组合进行评价。由于股票相对于无风险证券在短期内具有更大的波动性、暂时性，损失概率更高，投资者的短视性损失厌恶降低了股票对其的吸引力。因此，投资者持有股票必须要求比无风险证券高得多的风险补偿。

专栏 6.1　股权溢价之谜研究的后续进展

McGrattan 和 Prescott(2003)重新审视了 Mehra 和 Prescott(1985)的文章，提出其衡量股权溢价之谜的数据选择存在以下四个问题：①采用 90 天国债作为债权回报率的衡量标准，这一方法与代表性个体在寿命期限内配置资产相比周期过短；②以 S&P500 指数为股权回报率，没有考虑税收的影响；③以 S&P500 指数为股权回报率，没有考虑股权多样化成本的影响；④样本包括 1935—1960 年，这期间为第二次世界大战与朝鲜战争时期，许多政策限制导致了股权与债权间回报率的较大差异，应当剔除。

在分析 Mehra 和 Prescott(1985)存在的问题的基础上，McGrattan 和 Prescott(2003)提出了对股权与债权回报率的新衡量方法：①对股权溢价之谜是否存在的检验中剔除 1935—1960 年的战争时期；②剔除税收与多样化成本对股权与债权回报率的影响；③采用长期高级债权回报率作为债权回报率的衡量标准。在新的样本数据下平均债权回报率约为 4%，股权回报率约为 5%，股权的适度溢价是可解释的，即股权溢价之谜是不存在的。

然而关于股权溢价之谜的研究，至今仍然存在争议。

6.2 封闭式基金折价之谜

封闭式基金折价，即封闭式基金的交易价格长期低于资产净值的现象。根据有效市场假设，基金是无法获得超额收益的，但其收益应该处于资产定价模型的合理数值范围内，而不应该出现交易价格严重低于资产净值的情况，这一现象就称为封闭式基金折价之谜。10只封闭式基金1998年4月30日—2002年9月27日平均折价率如表6.1所示。

表6.1 10只封闭式基金1998/4/30—2002/9/27平均折价率

基金名称	平均折价率/%
基金开元	5.01
基金金泰	4.57
基金兴华	3.87
基金安信	3.38
基金裕阳	5.15
基金普惠	10.52
基金泰和	8.39
基金同益	10.85
基金景宏	9.27
基金汉盛	10.90

资料来源：韩国文，任伟红．我国封闭式基金折价的实证分析及行为金融学解释[J].中国软科学，2004(4)：59-64.

封闭式基金折价现象广泛存在，如表6.1中10只基金均处于折价状态，平均折价率为7%左右。封闭式基金的发展过程大致分为四个过程，从基金的溢价发行开始，然后基金出现折价，紧接着基金的折价率开始波动进入深度折价状态，最后折价又开始缩小。而关于封闭式基金折价之谜的传统解释，最具代表性的有代理成本、资产流动性、基金业绩和资产利得税。

代理成本理论指出封闭式基金收取的管理费用本身就会对基金的交易价格造成损失，这正是导致基金出现折价的最主要原因。Boudreauk(1973)认为如果基金收取的管理费用过高，或者未来的管理无法达到投资者预期，就会使封闭式基金出现折价。

而资产流动性理论则认为，一方面，如果封闭式基金持有的股票流动性较低，则基金本身的市场价值会因为流动性低而产生折价；另一方面，如果封闭式基金大量持有某只股票，则考虑到其变现时的流动性风险，交易价格自然会有一定下跌，而封闭式基金的资产净值是按照持有的股票的市场价格计算的，因此封闭式基金的交易价格会低于其资产净值。

Malkiel(1977)试图从基金业绩角度解释封闭式基金折价之谜，他认为当市场对基金未来盈利能力预期较差时，封闭式基金就会出现折价。然而这一解释并不完全正确，一些业绩良好的基金仍然会有折价现象出现。

资产利得税理论认为，由于资产利得税的存在，投资者在卖出实现资本升值的封闭式基金时，需要缴纳资本利得税，而缴纳的这部分资本利得税从另外一种角度来说就是卖出

基金的实际利得减少,即交易价格受到损失,因此会出现折价现象。

尽管传统金融学对封闭式基金折价之谜作出了诸多解释,但是实证研究表明,这些因素并不能解释全部的问题,只是有限地解释了其中一些现象而已。因此一些研究从行为金融角度合理地解释了封闭式基金折价之谜。

第一种解释是有限套利。投资者在进行套利时会受到佣金、买卖差价等成本约束及基本面风险、噪声交易风险等风险约束,因此套利是有限的。从而当封闭式基金出现折价时,套利交易者进行套利活动,如买入封闭式基金的同时卖空封闭式基金的标的股票,并不能完全使封闭式基金的交易价格回归资产净值,封闭式基金仍然会存在折价现象。

第二种解释是噪声交易者的存在。De Long,Shleifer,Summers 和 Waldmann(1990)最早提出了噪声交易者模型(DSSW),将投资者分成理性投资者和非理性投资者(噪声交易者)。由于市场中噪声交易者的大量存在,理性投资者在进行套利时,不仅要面临基本面风险,还要面临噪声交易者的非理性预期行为带来的风险,理性投资者可能会放弃套利甚至加入噪声交易者行列,阻碍基金价值的回归。

第三种解释是投资者情绪理论。Lee,Shleifer 和塞勒(1991)提出了投资者情绪假说,基金价格会受到噪声交易者情绪的影响。在封闭式基金上市时,噪声交易者会受到基金宣传等的影响,对基金未来的表现持有积极乐观的情绪,因此基金发行时多会溢价。而噪声交易者的情绪是随机变化的,导致基金的折价时常波动。此外,在长期过程中,理性投资者还将面对无法分散的噪声交易者风险,因此理性投资者要求更低的基金价格以弥补这部分风险,从而基金会表现出长期的折价。

第四种解释是信息不对称。Grullon 和 Wang(2001)从信息不对称角度对封闭式基金折价之谜作出了合理的解释,他们将封闭式基金的投资者分为投机者、分散投资者、套利者和噪声交易者,并构建模型。当封闭式基金持有的资产中有信息的投机者数量与其拥有的信息质量的乘积多于封闭式基金中的有信息的投机者数量与其拥有的信息质量的乘积时,基金会产生折价,即封闭式基金和封闭式基金持有的资产这两种的信息不对称状况决定了封闭式基金是否会出现折价。

延伸阅读 6.1　封闭式基金折价率的应用

封闭式基金的折价率可以作为投资者情绪的代理变量,研究投资者情绪是否会影响普通股票的收益率。Doukas 和 Milonas(2004)选取 1997—2002 年希腊封闭式基金的数据,得出结论认为,投资者情绪并不影响普通股票的风险,投资者情绪对小公司股票和共同基金的收益没有影响。

封闭式基金折价率为何可作为投资者情绪的代理变量?有学者认为,封闭式基金的折价反映了个人投资者的预期,封闭式基金的折价率受到投资者情绪的影响,因此封闭式基金折价率可以作为投资者情绪的间接代理变量,但其有效性仍受到质疑。随着文本挖掘等领域的发展,越来越多的学者采取更直接的情绪变量代替了封闭式基金的折价率。

资料来源:DOUKAS J A,MILONAS N T. *Investor sentiment and the closed-end fund puzzle:out-of-sample evidence*[J]. *European financial management*,2004,10(2):235-266.

6.3 日历效应

日历效应是指证券市场中，股票的收益率随时间发生变化，出现非正常的收益，即在不同时间下，投资收益率存在系统性差异，随着时间推移发生周期性的变动。根据日历或者时间长短来划分，日历效应又分为周内效应、月份效应和假日效应。

1. 周内效应

周内效应，即股票收益率在周内各天随时间发生规律性变化。1930 年，Kelly 首次提出纽约股票市场周一收益率异常偏低。Crossfrank(1973)以 1953 年至 1970 年标准普尔 500 指数每日收盘数据为研究对象，对美国股票市场是否存在周内效应现象进行了实证研究。结果显示，在此期间标准普尔 500 指数平均收益率在周五达到最大，而在周一最小，即呈现显著为正的周五效应和显著为负的周一效应。French(1980)对 1953 年至 1977 年的标准普尔 500 指数每日收盘指数周内效应现象进行了实证检验，结果显示，纽约股票市场的确存在周内效应，其中周一的平均收益率最低，并且具有统计显著性。

早期周内效应的研究结果均显示发达国家的股票市场多存在周一效应，即股票市场周一的平均收益率明显低于其他交易日。而关于“周一效应”的解释五花八门，有的从信息因素角度解释，认为坏消息在周末到周一出现的频率更高，而好消息则一般均匀分布在周内各天，因此导致了周一的低收益率。有的认为周一是一周工作的开始，投资者情绪差，从而造成低收益。还有的认为一般周五的收盘价会被有意抬高，因为周五的收盘价对于下周投资者的投资心理和投资决策影响重大。

2. 月份效应

月份效应，即股票收益率在年内各月随时间发生规律性变化，某些月份具有异常高的收益率而某些月份具有异常低的收益率。目前普遍的实证研究发现，发达国家股票市场具有显著的“1 月效应”，即 1 月的股票收益率显著高于其他月份。此外，对于新兴股票市场，也有研究证明存在月份效应。Aggarwal 和 Rivoli(1989)对我国香港地区、新加坡、马来西亚和菲律宾这四个新兴股市的月份效应进行了检验，发现除菲律宾外，在其余三个市场上，1 月的收益明显高于其他月份。

一些学者试图从税收角度进行解释，投资者为了减少缴纳的税款，往往会在年尾卖出账面亏损的股票，然后在 1 月重新将套现的资金投入股票市场，买入看好的股票，带动股市上涨。然而这只能对类似美国这样税收年度始于 1 月的国家进行合理的解释，而其他一些国家虽然存在 1 月效应，但是其税收年度却各不相同。

卡尼曼和特沃斯基(1982)提出的解释是，1 月效应的存在源于诸多国家圣诞节或者新年期间赠送礼物的习俗，造成了对现金的特别需求，提出 1 月效应受到投资者季节性投资偏好的影响，在圣诞和新年期间，财产从年老的风险厌恶投资者转移到年轻的风险偏好投资者身上。这似乎也是对 1 月效应的一种合理解释。

3. 中国股市的日历效应

股票市场的日历效应是市场非有效的体现，诸多实证研究表明，多数工业国家股票市场和一些新兴股票市场均存在日历效应，中国股票市场也不例外。但是不同于美国等发达股票市场的日历效应，中国股票市场的日历效应有其独特性。

我国股票市场并不存在周一效应，而存在显著为负的周二效应和显著为正的周五效应。奉立城(2000)以 1992 年 6 月 1 日至 1998 年 6 月 30 日上证综指和深证成指每日的收盘价数据为研究对象，在对收益率模型进行回归时，首先利用普通最小二乘法，然后利用以普通最小二乘法所得的对剩余项标准差的估计值的倒数为权重的加权最小二乘法对回归模型重新进行估计(表 6.2)。研究发现，我国股票市场无论是上证综指还是深证成指都呈现正的周五效应和负的周二效应。

表 6.2　沪深两市 1992—1998 年周内各天平均收益率

股票市场	时间	周一	周二	周三	周四	周五
上海	1992—1998	−0.071%	−0.417%**	0.278%	0.014%	0.541%***
深圳	1992—1998	0.066%	−0.230%	0.145%	0.019%	0.300%**

资料来源：奉立城．中国股票市场的“周内效应”[J]. 经济研究，2000(11)：50-57.

注：** 表示在 5%显著性水平下显著；*** 表示在 1%显著性水平下显著。

作为证券市场异象之一，日历效应的存在对于有效市场假说提出了强大的挑战。有效市场假说作为现代金融理论的核心，认为价格能够迅速地反映所有信息，它解释了资本市场对证券定价的有效性。投资者利用股票市场上的信息所进行的交易，不可能存在非正常的收益，只能赚取风险调整后的平均市场收益率。然而大量证券市场异象的存在，使得投资者可以运用特殊的投资策略获得超额收益。

专栏 6.2　日历效应消失了吗?

随着周一效应被投资者所熟知，投资者套利行为使得股票市场的周一效应在慢慢消失，逐渐出现其他的周内效应。Mehdian 和 Perry(2001)认为 1987 年之后美国股票市场的周一效应就不存在了。这也说明当某种日历效应一旦被较多地发现并认同，就会渐渐消失。

对于中国股票市场早期存在的周五效应，丁荣余和张兵(2005)认为奉立城(2000)没有对数据分组来检验日历效应是否随时间推移而消失，他们运用滚动样本检验和分年度检验发现从 1997 年开始中国股票市场的周五效应就已经消失了。

资料来源：MEHDIAN S，PERRY M J. *The reversal of the monday effect：new evidence from US equity markets* [J]. *Journal of business finance & accounting*，2001，28(7-8)：1043-1065.

丁荣余，张兵．中国证券市场星期效应逐渐消失的经验证据[J]. 管理工程学报，2005，19(3)：145-150.

陆磊和刘思峰(2008)在研究中国股市的节日效应时发现中国股市同时存在节前效应和节后效应。此外，研究表明在考虑其他日历效应后，节日效应依然显著为正，这说明中国股市节日前后的异常收益现象是由节日效应引发的，而不是其他日历效应(如周一效应、周五效应和 1 月效应等)。

6.4 规模效应

规模效应是指股票的平均收益率与公司规模之间存在规律性的负相关关系(图 6.3),尤其是一些小市值公司股票收益率超过市场平均水平的现象,因此规模效应又称小公司效应。

Banz(1981)将纽约证券交易所上市交易的股票按市值大小分为五组,研究股票市值与收益率的关系,发现市值最小的组的平均收益率高于其他组,因此首次提出了小公司效应。他还发现规模效应和公司市值的大小并不是线性关系,规模效应仅发生在小市值的公司中,而中等市值和大市值的公司之间并没有明显的区别。Fama 和 French 在三因子模型的一系列研究中也证明了公司规模对股票收益率影响显著。

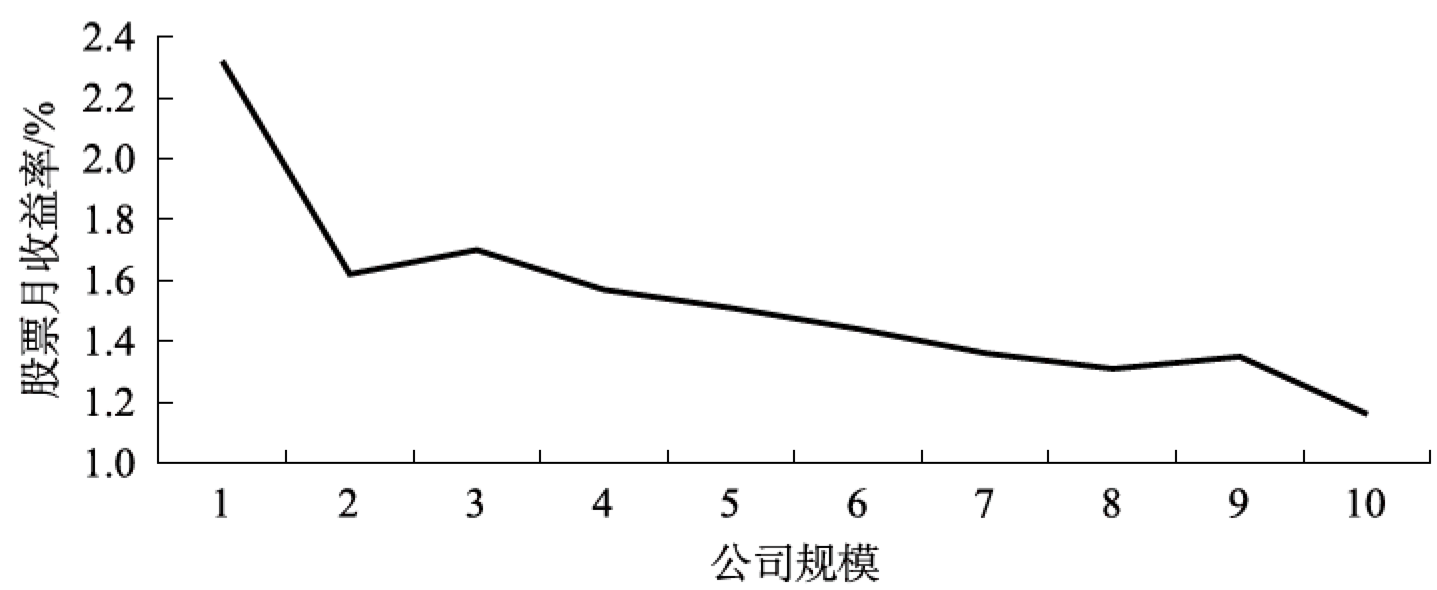


图 6.3 1976—1995 年美国上市公司不同公司规模股票平均月收益率

注:1～10 公司规模递增。

资料来源:KIM M K,BURNIE,D A. *The firm size effect and the economic cycle*[J]. *Journal of financial research*,2002,25(1),111-124.

从图 6.3 可见,规模效应确实存在,股票收益率与公司规模存在显著的负向关系。规模效应的解释有许多,主要的原因有以下几点。

1. *风险假说*

在资本市场异象的研究中,有的研究者利用风险对相关异象进行解释,这种解释试图捍卫有效市场理论的权威性。他们以多因素资产定价模型为基础。在规模效应研究领域,风险假说认为规模效应的存在并不否认市场有效假说,而是由于没有考虑系统性风险,导致把正常股票收益误判为超额收益。

2. *样本选择偏差*

一些学者认为规模效应的存在是因为在数据挖掘时,实际上并不存在表明收益率与那些解释变量之间存在关系的证据。并且不同学者的研究样本不同,例如不是所有的样本都包括发生了财务危机的公司等,样本选择偏差的存在使得实证结果会出现公司规模与股票收益相关或者不相关的现象。

3. *流动性效应*

小公司交易相较于大公司更加不频繁,使得小公司流动性更低,为了弥补流动性低带

来的可能存在的变现损失，小公司股票要求更高的收益。有些学者在研究股票流动性时还发现流动性往往与公司的规模、账面市值比等因素存在显著的相关性，如高账面市值比和小公司常常有较低的盈利性，更容易存在管理不善等问题。而公司管理不善等往往会使公司流动性变差，因此流动性因素与公司的收益之间存在一定的间接联系。

4. 行为金融理论

Baker 和 Wurgler(2006)等学者试图从投资者情绪理论角度寻找规模效应的产生原因，他们认为小公司的股票比大公司的股票更容易受到市场情绪的影响，因此噪声交易者制造的高情绪风险需要更高的收益作为补偿。有学者在 Fama 和 French 三因子模型的基础上将澳大利亚股市作为研究对象，认为规模效应主要是由情绪因素引起的。

还有一个解释是“忽略效应”。在股票市场中，机构投资者更加具备操纵市场的力量，但机构投资者往往只研究大公司，却很少关注和研究小公司，所以投资者对于小公司的生产经营和盈利状况知之甚少，产生了信息不对称问题。而信息的不对称使风险提高了，因此一些学者认为小公司信息不对称带来的风险补偿是造成小公司超额收益率更高的原因。

此外，规模效应的解释还有“窗帘效应”。有些研究发现，规模效应也存在时间特征，往往 1 月时规模效应较为显著。除了一定的避税因素外，还有就是机构投资者，如基金经理为了粉饰报表，避免一些业绩差的股票出现在年报中，经常在年底抛售业绩差的股票，而在第二年反向买进，使股票价格上涨。

汪炜和周宇(2002)以上证市场为对象考察我国股票市场的规模效应，研究发现在 1997—2001 年，小公司资产组合每年 12 个月中有 9 个月的平均相对收益率高于上证综合指数收益率，规模效应即小公司效应表现显著。他们认为，小公司股价存在的连续超额收益率源自我国股市复杂市场结构和制度背景下的小公司股票流动性问题，是对高额隐性交易成本的补偿。

6.5 账面市值比效应

账面市值比效应，又称 BM 效应，是指公司账面价值和市场价值之比与公司收益率成正比，即账面市值比高的公司的平均收益率高于账面市值比低的公司。账面市值比效应存在于全球各个主要的证券市场中。

有效市场假说对账面市值比效应从风险角度作出解释，认为高账面市值比的公司有更高的破产风险，遭遇财务危机的可能性更高，因此这些公司的股票需要更高的风险补偿，即高账面市值比公司的收益其实仍然是一种风险的补偿。然而这一解释并不十分站得住脚，高破产风险未必就代表高收益。

行为金融理论从投资者心理预期、错误定价等角度提出了对账面市值比效应的不同解释。

账面市值比效应的存在是市场对公司基本面过度反应的结果。投资者对高账面市值比公司基本面的坏消息过度悲观，而对低账面市值比公司基本面的好消息过度乐观，因此

低估高账面市值比公司而高估低账面市值比公司。经过一段时间理性回归之后，两种公司的收益发生反转，使得高账面市值比公司比低账面市值比公司有更高的收益。然而这一解释也存在一点不足，公司的基本面价值对投资者来说是难以准确衡量的，且不同投资者的主观判断多有不同。

一些学者认为投资者对高账面市值比公司和低账面市值比公司不同未来盈余的预期会影响账面市值比效应。投资者对高账面市值比公司的预期有更高的未来盈余，即并不是投资者对公司基本面价值的过度反应造成了账面市值比效应，而是由于投资者对两种账面市值比公司的不同未来盈余的主观预期产生了账面市值比效应。

迄今为止，账面市值比效应仍然让研究证券市场异象的学者争论不一，未来需要更完整的理论解释和实证的经验支持。

6.6 投资者关注度

Merton(1987)首次提出投资者关注度，认为投资者关注度通过影响投资者的行为而进一步引起市场的波动。此后随着行为金融学的发展，投资者关注度已成为研究市场有效性的重要主题。

投资者关注分为关注和有限关注。关注是指投资者因特定的引人关注的事件而对相应股票产生偏离基本面的过度反应。而有限关注是指由于投资者时间和精力有限，不可能考虑所有的股票，对于影响股票基本面的信息反应不足，从而对股票市场的价格和成交量产生影响的情况关注有限(Engelberg 等，2009)[①]。

股票市场有那么多股票，投资者要选择哪些呢？因此投资者关注是一种稀有资源，那些吸引了大量关注的股票更有可能被投资者纳入考虑范围。投资者对于吸引其关注的股票信息进行分析判断，并依此调整自己的行为，从而引起股票市场的波动，如股价的偏离和交易量的异常等。此外，研究发现，投资者往往会买入自己有限关注的股票，使被关注的股票有一个短暂性的股价上升，即那些获得投资者关注的股票往往会有超额收益。后续的研究也发现投资者关注度对股票当日收益率有显著的正向影响，但之后影响将会反转。

专栏6.3 投资者关注度的指标

投资者关注度的指标衡量主要有三种方式：一是利用金融资产本身的交易特性和价格行为间接衡量，如交易量、超额收益、市场指数累计收益等；二是利用新闻媒体报道及广告支出间接衡量；三是利用搜索引擎的趋势统计功能直接衡量，如一些搜索引擎的搜索指数等(表6.3)。

---

① 宋双杰，曹晖，杨坤．投资者关注与IPO异象——来自网络搜索量的经验证据[J]．经济研究，2011(s1)：145-155.

表 6.3　投资者关注度指标衡量方式

作者	投资者关注度的指标
Barber 和 Odean，2008	异常交易量(AV)：股票 i 在第 t 天的异常交易量。 $AV_{it}=V_{it}/\bar{V}_{it}$，其中 V_{it} 是股票 i 在第 t 天的美元成交量，$\bar{V}_{it}$ 是股票第 t 天前 252 个交易日平均交易量。 再基于异常交易量构造关于交易不平衡(买方成交量和卖方成交量差)的日时间序列
饶育蕾，彭叠峰，成大超，2010	媒体关注度(AT)：上市公司 i 在第 t 期被关注的程度。 $AT_{i,t}=(N_t-rank_{i,t})/N_t$，其中 N_t 表示在 t 期上市公司总家数，$rank_{i,t}$ 表示上市公司 i 在 t 期的新闻条数在当期所有上市公司中的排名
宋双杰，曹晖，杨坤，2011	异常搜索量(ASVI)：谷歌趋势每只股票每周搜索量指数。 $ASVI_t=\log(SVI_t)-\log[Med(SVI_{t-1},\cdots,SVI_{t-8})]$ 其中，$\log(SVI_t)$ 是第 t 周搜索量的对数，$\log[Med(SVI_{t-1},\cdots,SVI_{t-8})]$ 是第 $t-8$ 周到 $t-1$ 周每周搜索量的中位数的对数

Barber 和 Odean(2008)研究了投资者关注对个人投资者和机构投资者买入行为的影响。他们为判断投资者是否对一家公司关注建立了三个标准：①股票有异常的日交易量；②股票前一天的日收益率；③公司在当天的报纸中是否出现。此外，他们将四种投资者的交易行为作为样本：①账户所在经纪公司佣金有大折扣的投资者；②账户所在经纪公司佣金折扣小但经纪公司宣传其交易执行质量的投资者；③账户所在经纪公司是大型零售经纪公司的交易者；④专业资金经理人。研究结果表明，个人投资者往往倾向于买入吸引其关注的股票。但是个人投资者并不会买入每一只吸引其关注的股票，在投资者关注的股票中决定买入哪些股票取决于投资者个人的偏好，如逆向投资者会购买受到投资者关注但是却不被其他投资者看好的股票。而机构投资者却不满足于只买入投资者关注度高的股票。机构投资者有更多的时间和精力去持续关注更广范围的股票，并且有着更为明确的买入标准。

在投资者关注度的研究中关注度的衡量一直是研究的难点之一。早期投资者关注度的代理变量主要有交易量、媒体报道、换手率、广告支出等。虽然这些代理变量与投资者关注度有一定的联系，但并不能直接反映投资者关注度的分布与强度。因此，近年来涌现出一些更为直接的衡量投资者关注度的变量。如宋双杰等(2011)利用谷歌趋势提供的网络搜索量数据构建投资者关注度的直接衡量指标，发现投资者关注对资产价格有显著的影响，IPO(首次公开募股)前个股的网络搜索量对于市场热销程度、首日超额收益和长期表现有更好的解释力与预测力。俞庆进和张兵(2012)采用百度指数作为关注度的指标，验证了投资者关注对股票市场影响的存在性，且能给股票价格带来正向影响，但很快会反转。

6.7　本地偏好

传统的金融投资组合理论认为，投资组合可以有效地降低非系统性风险，投资者应该

最优化投资组合，在更广的范围内合理配置资产。然而在现实中，存在与此违背的市场异象，投资者在选择投资组合时往往表现出明显的本地偏好(home bias and local bias)。

本地偏好是行为金融理论研究中人们发现的一种重要的投资者行为现象，即机构或个人投资者更愿意投资于本国股票市场、熟悉的或就职的公司、离自己住地近的公司等。本地偏好有两个方面：一方面是指投资者的资产配置中绝大部分选择在本国资本市场(home bias)，French 和 Porterba(1991)发现美国投资者将他们 94%的资金投资于本国股票市场，尽管美国股票市场份额仅占全球股票市场的 48%。而其他国家也存在相同的情况。另一方面是在自己国家，投资者也往往更愿意投资熟悉的或就职的公司或者地理距离更近的公司(local bias)。这两种本地偏好存在一定的区别，因为全球金融市场中国家间的本地偏好会受到国家资本流动监管限制等因素的影响，尽管有研究表明这种本地偏好同样存在于限制境外证券投资的国家和不限制境外证券投资的国家。

行为金融从不同角度对本地偏好现象提出了合理的解释。

1. 信息不对称

一些学者认为，投资者投资本地股票是因为他们对本地公司更具有信息优势，投资者收集信息的成本更低。Coval 和 Moskowitz(1999)研究了共同基金的收益与其投资的公司的地理距离之间的关系，发现本地投资的共同基金有显著更高的超额收益。他们认为，本地投资者能够利用地理优势获得本地公司更真实的信息，这种本地投资者和非本地投资者之间的信息不对称使得投资者更加偏好本地投资。

2. 熟悉效应

投资者之所以会投资本地公司，有一部分是因为他们对这些公司更加熟悉。即使投资者可以花同样的时间和精力来研究非本地公司以获得同样多的信息，但由于熟悉效应或者对本地的热爱，他们认为自己对本地公司拥有更多的信息，而更偏好投资本地公司。

杨晓兰等(2016)以投资者在东方财富网股吧针对创业板上市公司发表的帖子为研究对象，通过帖子 IP(互联网协议地址)地址识别构建本地关注指标，发现本地关注对股票收益率的影响取决于投资者情绪，当投资者持积极情绪时，本地关注对股票收益率有显著的正向影响；当投资者持消极情绪时，该影响显著为负。

延伸阅读 6.2 城市中唯一游戏效应

Hong 等(2008)提出了城市中唯一游戏效应(only-game-in-town effect)，支持了本地偏好。他们认为股票价格和商品、服务一样受到供求关系的影响，正如城市中还有许多空房间的旅馆房价会便宜一点一样，当本地公司必须和其他附近的公司竞争得到投资者资金的时候，它的股票价格会更低一点。因此，如果存在本地关注度，随着本地公司总账面总市值与本地投资者总风险容忍程度之比下降，股票价格会降低。所以在人口密度较低的地区，这种城市中唯一游戏效应就会特别明显。

资料来源：HONG H G，KUBIK J D，Stein J C. *The only game in town*：*stock-price consequences of local bias*[J]. *Journal of financial economics*，2008，90(1)：20-37.

6.8 无信息反应效应

相比于上市公司兼并收购、资产重组、股利发放等向股票市场传递公司相关信息的事件,股票市场上也存在一些“消息真空”事件,这些事件与上市公司的基础价值无关,但是却引起股票价格的显著变化,如更名效应和指数效应。

中国上市公司更名现象并不少见,上市公司因发生股权转让、兼并收购和资产置换等活动使得股东、主营业务发生变化,往往伴随公司名称和股票名称的改变。由于股票更名前上市公司重要经济事件均已公告过,理论上股票名称变更公告的发布除了宣告公司股票名称发生变化外,并不包含任何未能预期的信息,即有效市场中股票更名公告不会给上市公司的股价和交易量带来显著的影响。然而,我国股市股票更名公告往往会导致短期内显著的股价飞跃,即更名效应。这可能是众多上市公司热衷于改名的重要原因之一。刘力和田雅静(2004)使用1999年到2001年发生股票名称变更的148家上市公司为样本,发现股票更名公告会引起显著的股价上涨,产生正的异常收益,随后股价又反转下跌,产生负的异常收益。邓建平和曾勇(2006)采用事件研究法,发现股票更名前后存在超常交易量和超额收益率,且市场对更名事件存在过度反应,这一结论与刘力和田雅静(2004)的研究结论一致。此外,他们深入研究了不同股票更名原因的影响差异,发现投资者对第一大股东更换导致的股票更名反应强于主营业务更换导致的股票更名,并确认了我国股市较严重的题材炒作和投机现象。

指数效应是指在成分指数调整前后,调入指数和调出指数的股票的价格与成交量存在异常的市场反应。指数效应广泛存在于各国股票市场。史莱佛(1986)发现从1976年到1983年,调入标普500指数的股票存在2.79%的异常超额收益。日经225指数(Liu,2000)和加拿大多伦多股票交易300指数(Kaul等,2000)已表现出显著的指数效应。指数效应的相关研究众多,国内刘斌才(2004)发现上证180指数中第一次调出指数的股票在调整日存在显著为负的异常收益率。蒋金锋(2008)运用事件研究法对沪深300指数从2005年6月到2007年12月的历次样本股调整事件中的市场反应做了详细的研究,发现沪深300指数的指数效应显著存在,调入指数股票在事件当日的异常收益率显著为正,而调出指数股票在事件当日的异常收益率显著为负。此外,研究还表明小公司被调入沪深300指数所产生的指数效应要显著大于大公司被调入沪深300指数所产生的指数效应。

本章小结

(1)异象是明显违背有效市场假说的经验现象,主要有股权溢价之谜、封闭式基金折价之谜、日历效应、规模效应、账面市值比效应、投资者关注度、本地偏好和无信息反应效应等。

(2)股权溢价是指风险证券的收益高出无风险债券收益的部分。

(3) 封闭式基金折价,即封闭式基金的交易价格长期低于资产净值的现象。

(4) 日历效应是指证券市场中,股票的收益率随时间发生变化,出现非正常的收益,即在不同时间下,投资收益率存在系统性差异,随着时间推移发生周期性的变动。

(5) 规模效应是指股票的平均收益率与公司规模之间存在规律性的负相关关系,尤其是一些小市值公司股票收益率超过市场平均水平的现象,因此规模效应又称小公司效应。

(6) 账面市值比效应,又称 BM 效应,是指公司账面价值和市场价值之比与公司收益率成正比,即账面市值比高的公司的平均收益率高于账面市值比低的公司。

(7) 投资者关注分为关注和有限关注。关注是指投资者因特定的引人关注的事件而对相应股票产生偏离基本面的过度反应。而有限关注是指由于投资者时间和精力有限,不可能考虑所有的股票,对于影响股票基本面的信息反映不足,从而对股票市场的价格和成交量产生影响的情况关注有限。

(8) 本地偏好是指机构或个人投资者更愿意投资于本国股票市场、熟悉的或就职的公司、离自己住地近的公司等。

(9) 指数效应是指在成分指数调整前后,调入指数和调出指数的股票的价格与成交量存在异常的市场反应。

思考题

1. 名词解释

(1) 股权溢价。

(2) 规模效应。

(3) 投资者关注度。

(4) 本地偏好。

2. 通过查阅资料和文献,列举直接衡量投资者关注度的其他方法。

3. 描述一个你知道的证券市场异象,并解释这一异象出现的原因。

4. 恒生 AH 股溢价指数是由香港恒生指数服务公司于 2007 年 7 月 9 日正式对外发布,该指数追踪在内地和香港两地同时上市的股票(内地称 A 股,香港称 H 股)的价格差异。该指数根据纳入指数计算的成分股的 A 股及 H 股的流通市值,计算出 A 股相对 H 股的加权平均溢价。指数越高,代表 A 股相对 H 股溢价越高;反之,代表 A 股相对 H 股越便宜。

请收集恒生 AH 股溢价指数,并联系所学过的知识,解释为什么会出现 A 股和 H 股的溢价。

第三部分

行为金融学应用



第7章

过度自信

引导案例:股市真的可预测吗

2013年诺贝尔经济学奖获得者席勒在他的著作《非理性繁荣》中曾记录一个很有趣的小实验。在1987年10月19日美国股市大跌之后,他对投资者进行过一个调查。结果显示,在当天购买股票的个人投资者和机构投资者中,分别有47.1%和47.9%的人表示自己知道10月19日什么时候股市会发生反弹。即使是在当天没有交易的个人投资者和机构投资者中,也有29.2%和28.0%的人分别给出了相同的答案。显然,这些人的答案是存在问题的。股市具有不可预测的特性,但是这些人又没有撒谎,为什么那么多人表示自己能够预见股市当天的变化呢?席勒进一步询问了这些投资者为什么他们能够知道什么时候发生反弹。大多数人的回答与"心理学""历史证据"或"直觉"有关,却很少有人能够说出理论和实际支撑,因此他们所表现的很可能只是一种心理偏差。这样的例子还有很多:在2008年经济危机发生后,很多人都表示自己早就"知道"这个危机不可避免;人们总是认为自己入手的股票会涨而出手的股票会跌;分析师和专家对股市的预测并不比普通人更加精确……这些情况都与一种心理特征有关——过度自信。

从心理学来说,人们普遍趋向于过度自信并且难以消除,这一特点也在经济学和行为金融学中得到应用。过度自信是指对自己的知识和能力有过高的估计,过度重视自己所拥有的信息而忽略市场上或他人的信息,或者认为自己能够掌控未来的事件。

本章的第一部分将介绍过度自信的四种不同的表现形式:错误校准、自我感觉良好效应、控制幻觉和过度乐观,以及其相应的测度方法。三类心理偏差——自归因偏差、后视偏差和确认性偏差导致了过度自信的产生并阻碍过度自信的纠正。

本章的第二部分将介绍过度自信是如何对投资者的金融决策产生影响的。过度自信会在投资者的判断、估值和投资决策阶段产生不同的影响,并且过度自信会导致投资者过度交易和采取激进的投资策略,因此使得组合的收益降低。既然过度自信给投资带来这

么多不利之处，为什么过度自信的投资者还能广泛地存在于金融市场中？本章将对此给出一定的解释。

本章的第三部分主要介绍过度自信在公司金融中的应用。首先解释为什么公司管理层更容易表现出过度自信的特征，接下来进一步利用管理层的过度自信来解释公司的投资扭曲现象，并探讨过度自信对兼并收购、公司资本结构以及股利分发政策的影响。

7.1 过度自信的表现形式与测量方法

和人类心理的复杂性一样，过度自信作为一种心理偏差，有着多种表现形式，从其不同的表现形式出发，过度自信也有不同的测度方法。然而，由于过度自信的复杂性和多变性，至今心理学以及行为金融学界对其仍没有一个统一、权威的测度方法。下面将要介绍的过度自信的四种表现形式和测度方法也只是众多分类中较为经典和简单的方法。本章的实验指引中也具体介绍了如何在实验研究中应用这些方法。

1. 错误校准

错误校准(mis-calibration)表现为个体对自身能力和知识的高估，从而产生对个人决策准确性过高的评价，通常通过校准实验(calibration test)来进行测量。校准实验有多种形式，其中一种经常被使用的是置信区间问题任务法。在实验中，实验被试通常被要求就一些常识性的问题给出90%的置信区间(90%置信区间的含义是实验被试认为正确的值有90%的可能性落入他所给出的区间)，实验被试回答的置信区间若包含正确答案，即认为实验被试回答正确。例如，珠穆朗玛峰的海拔高度是多少米？精确的答案是8 844米。假设第一个实验被试清楚地记得珠穆朗玛峰的高度是8 000多米，那么他可以给出下限为8 000、上限为9 000的区间。另一个实验被试则认识到自己对此问题不清楚，他给出下限5 000、上限10 000的区间。这两个实验被试都属于回答正确，因为正确答案在他们给出的区间内。假设第三个实验被试对自己的地理知识非常自信，他给出一个更小的区间：下限8 800，上限8 840，不幸的是这个区间是错误的，并没有包含8 844这个正确答案。可见实验被试越是对自身能力和知识缺乏正确的评价，就越可能给出错误的区间。当实验被试回答完全部问题之后，可以计算出他回答问题的正确率。实验被试的错误校准程度为1减去他的正确率，数值越大，意味着正确率越低，实验被试的过度自信水平则越高。根据90%的置信区间可知，如果实验被试的错误校准程度为0.1，则他的自信水平是适当的，既没有表现出过度自信，也没有表现出自信不足；若错误校准程度大于0.1，则说明他是过度自信的。这是一种客观概率测量法，因为实验被试对答案的自信程度已经被客观地定为置信度90%。

理论上，如果同一个实验被试回答足够多的(90%)区间估计问题或者有足够多的实验被试回答一定数量的(90%)区间估计问题，排除样本误差之后，其整体正确率应该接近90%。但是，在很多实验中，实验被试的正确率都低于设定的置信概率，实验被试都表现出过度自信，也就是说，实验被试通常给出一个过于狭窄的置信区间。例如，用上述方法测试110名在校大学生(44名男性，66名女性)的过度自信程度，发现他们错误校准程度

即过度自信程度的均值是0.66,即回答问题的准确率平均只有34%。

对错误校准程度的测度方法还有其他形式。其中一种测量方法涉及实验被试对自身答案自信程度的估计,称之为主观概率估计法(probabilistic judgment)。在实验中,实验被试首先被要求对一些难度适中并且具有唯一正确数字答案的问题给出答案的区间估计,随后他们需要对自己的每一个答案给出自信程度,即自己主观认为答案正确的可能性大小。过度自信的程度等于其自我评估的平均正确率与其实际答题的正确率相减(过度自信水平=实验被试认为自己选择正确的平均概率—实验被试选择的实际正确率)。若该值为负,则说明实验被试不够自信,若该值为正,则说明他过度自信。主观概率测度法被很多实验设计采用,但是和客观概率法相比,其测度的过度自信水平通常较低。另一种测量方法与主观概率估计法类似,叫作频率估计法(frequency judgment),实验被试在答题结束后需要估计自己答题正确的个数,通过答题正确比例和预估正确比例的比较,确定实验被试的过度自信程度。如果实验被试认为自己能答对70%的题目,但实际上只答对了50%的题目,就说明他是过度自信的。

除了置信区间问题任务法这一种题目形式之外,研究者也会采用二择一任务法(二选一的选择题)和分类任务法(例如:实验被试根据特定的症状表现把假定的患者按疾病类型划分,并估计患者患该种疾病的概率,通过对患病概率的估计测量过度自信的程度)①。

值得注意的是,问题的设置对测试结果的影响很大。实验被试通常对难度大的问题更容易表现出过度自信而对简单的问题表现出自信不足。事实上,区间估计问题的提出形式(是选择题还是问答题)、给定置信度的高低、问题涉及的领域都会影响结果。

2. 自我感觉良好效应

自我感觉良好效应(better-than-average effect)涉及与他人的比较,根据自我感觉良好效应,通常人们会认为自己比平均水平更加聪明或者优秀。但是实际上,只有50%的人能够超过平均水平,因此这也是过度自信的一个表现形式。在Svenson(1981)的一个著名实验中,他发现93%的美国司机和69%的瑞典司机都认为自己的车技比平均水平高。但这显然是不可能的,事实上生活中这样的例子比比皆是,人们自我感觉良好的现象普遍存在。

那么自我感觉良好效应应该如何测定呢?通常的做法是,要求实验被试对自己的能力、知识或者表现和平均水平进行比较,通常由一个或者几个题目组成。例如,要求实验被试对其驾驶技术进行评分,要求实验被试对自己在朋友中的文化水平进行评分。也有测试在股票交易实验之前进行,要求实验被试对自己在这次股票交易实验中的收益排名

---

① 如需进一步了解这几种不同的题目形式的读者可以参考以下文献:

GIGERENZER G, HOFFRAGE U, KLEINBOLTING H. *Probabilistic mental models: a brunswikian theory of confidence*[J]. *Psychological review*, 1991, 98(4): 506-528.

LIBERMAN V. *Local and global judgments of confidence*[J]. *Journal of experimental psychology: learning, memory and cognition*, 2004, 30(3): 729-732.

SIECK W R, YATES J F. *Overconfidence effects in category learning: a comparison of connectionist and exemplar memory models*[J]. *Journal of experimental psychology: learning, memory and cognition*, 2001, 27(4): 1003-1021.

进行预期。若实验被试的回答显示其认为自己处于平均水平，则说明其自信水平处于标准水平；若实验被试认为自己的排名越高，其自我感觉良好程度越高。具体的测试题目可以参考本章末的实验指引。

为什么自我感觉良好效应如此普遍地存在？造成人们自我感觉良好的一个原因是人们对优秀的定义是不明确的。怎样才是优秀的，怎样才是最好的，每个人都有自己的标准，而人们心中的定义都会倾向于提高自己的排名。从动机层面考虑，自我感觉良好可以使自己更加自尊自信；从认知角度考虑，对大多数人来说，最容易想到的标准总是那个能够让自己显得最优秀的标准。

3. 控制幻觉

控制幻觉(illusion of control)主要是指过高预期了利于自身情况发生的可能性，从而产生一种能够对周围事物进行控制的错觉。举例来说，赌博者总会认为骰子的点数或者牌的大小是他们可以控制的，在希望点数大的时候，他们会更加用力地掷骰子，而当他们希望点数小的时候，他们掷骰子会更轻；司机总会认为当自己驾驶时，发生车祸的概率会比较小。

经济学家已经做过很多与控制幻觉相关的实验。在一个实验中，实验被试需要预测30轮硬币投掷的结果，研究者通过实验控制使得每一个实验被试都能预测对一半的结果(每个实验被试预测对的轮数分布不同)。随后，实验被试会获得自己预测结果的回馈，一部分的人被告知他们前面几轮的预测都是正确的，而另一部分的实验被试则被告知他们预测正确的轮数平均地分布于30轮投掷中。结果发现，第一组实验被试高估了自己预测对的总轮数，并且对自己以后的表现也有了更高的预期。这个实验结果可以用首因效应来解释，人们对较早获得的信息给予更高的比重。进一步的研究发现，40%的人认为他们预测硬币投掷结果的成功率会随着练习的增多而提高，25%的人认为在预测过程中的分心会使得成功率降低，但事实上预测硬币投掷结果是一个纯粹的运气游戏。在另一个实验中，一部分实验被试获得随机分配的彩票，而另一部分人则可以自己选择彩票号码。研究发现，即使能够获得一张有更大中奖概率的彩票，自己选择彩票号码的实验被试更加不愿意和他人交换彩票。

控制幻觉在人们面对随机事件时表现得更加淋漓尽致，特别是当随机事件涉及越多与"技巧"相关的内容时，人们更容易产生控制幻觉。人们产生控制幻觉遵循两个步骤：希望控制结果的意图和与随机事件结果相关的行为。虽然事件结果是随机的，但是引发结果的行为会诱导人们认为自己在一定程度上控制了结果。当人们面对自己不能控制的不确定事件时，则会更倾向于产生控制幻觉，事件的可控性越低，这种感觉越强烈。

对控制幻觉的衡量与自我感觉良好效应类似，可以通过让实验被试对几个陈述(问题)进行评分来确定。在不少知名的实验中用到过以下几个问题：你认为你的估值是准确的；你认为你能够很容易地专注于目标并达到目标；对于发生的任何问题，我都能靠自己解决，等等。

4. 过度乐观

Taylor 和 Brown(1988)曾指出有些个体相信"未来会更好，特别是对我自己"，也就是

说，他们相信幸运的事情发生在自己身上的可能性高于发生在别人身上。这就是过度自信的一种表现形式——过度乐观（excessive optimistic），它与控制幻觉紧密相关，通常表现为人们对未来不切实际的预期。过度乐观的人会对自己希望发生的事情赋予过高的发生概率，而对自己不希望发生的事件赋予过低的概率。这样的例子很多，如人们高估自己中彩票的概率而低估自己得癌症的概率，学生总是认为他们的分数被评判得过低，与他们的预期有差距。过度乐观也经常发生在计划制订时。人们不能按时完成自己计划的情况时常发生，这就是因为他们对自己的执行能力过度乐观以至于制订了不切实际的计划。另外，过度乐观也会导致完成成本的增加，预算超支对于企业和大型工程来说是非常常见的，究其原因，也是因为制订了过于乐观的计划。有外媒统计出全球预算超支最严重巨型工程名单，其中英吉利海峡隧道居首，超出预算高达 211 亿美元，而中国的三峡大坝工程紧跟其后，超出预算约 161 亿美元，位列第三位的波士顿大隧道至今仍有些收尾工程在进行中，超出预算 134 亿美元。

5. 影响过度自信的人口学因素

虽然人类普遍存在过度自信的特征，但不同人之间的过度自信程度有显著的差异。

男性和女性在过度自信之间的差别是非常显著的。很多研究表明，男性比女性更加表现出过度自信的特质。和女性相比，男性更加相信自己的判断，在金融市场上进行更多的交易，认为自己能够获得更高的收益。在一个实验中，当男性和女性同时被提问以下两个问题：你认为在未来的 12 个月中，你的投资组合收益将会是多少；你认为在未来的 12 个月中，股票市场的平均收益会是多少。虽然男性和女性都认为自己的投资组合会击败市场组合，但是男性期望的超额收益（2.8%）远高于女性期望的超额收益（2.1%），这说明从自我感觉良好效应的角度来说，男性和女性都存在过度自信，但是男性比女性程度更深。

和普通人一样，专业人士在他们的专业领域也受到过度自信的影响，专业人士的过度自信程度甚至可能超过非专业人士，在金融领域尤甚。特沃斯基曾经指出，在股票市场这样一个可预测性很低的领域，专业人士会比普通人更倾向于过度自信，因为相较于缺乏专业知识的普通人，专业人士掌握了经济理论和模型，他们会过高地估计理论和模型的作用。他发现，投资决策速度越快，投资者对这个决策的自信程度就越高，而在金融市场上，特别是股票市场、期货市场等，行情变化非常快，参与者的决策时间往往是很短的，这就更导致了金融市场参与者的过度自信。

金融市场的参与者总是希望自己能够找到被低估的资产，获得高于市场的收益，但是事实上，获得超额收益是困难的。在复杂的任务情境下，人们更容易表现出过度自信，因此金融市场参与者更普遍地存在过度自信的心理偏差。选择性偏误（selection bias）和幸存者偏差（survivorship bias）使这一现象更加明显。选择性偏误使积极参与金融市场的人更加过度自信。这是因为对自己的知识和能力有信心的人更可能选择积极参与资本市场，从事金融工作。而这些人当中，一部分人的能力和自己的认知是相符的，而另一部分人则过于高估自己的能力（也就是过度自信者）。另外，幸存者偏差使得成功的交易者留在金融市场（失败的交易者更有可能离开金融市场），掌握更多的资金，并因此而变得越来

越过度自信。投资的成功可能会让投资者将成功原因更多地归因于自我能力而非运气，从而使他们变得更加的过度自信。虽然随着经历的增多，投资者的自我评价会更加准确，从而可能减弱过度自信，但是在一个不断有新的投资者加入的金融市场，过度自信的投资者仍旧会源源不断地出现。

虽然过度自信会引起很多不必要的麻烦，带来不必要的成本增加，但是在一些情况下，过度自信也有有利的方面。在个人行为可以对事件结果产生影响的情况下，乐观的想法能够促使人们坚持完成任务而不是中途放弃。特别是对于比较长远的目标或者已经付出了很多努力的目标，人们会有更加乐观的预期，而研究证明此时过度自信有利于完成目标。

在现实生活中，人们通常不可能只表现出一种形式的过度自信，经常多种形式一起发生，难以严格地进行区分。并且，衡量过度自信程度的方法和标准多种多样，而通过不同的方法衡量的过度自信程度往往是不一致的，目前为止还没有一个权威的统一标准。

专栏 7.1　多种形式过度自信之间的不一致

在 Glaser 和 Weber(2005)的研究中，他们通过在线问卷对来自德国一家在线经纪公司的 2 015 位投资者的过度自信情况进行了调查。他们考虑了过度自信的多种不同形式，并分别利用不同的方法衡量过度自信水平，总共获得了 7 种能够代表过度自信水平的指标。MICA 表示通过置信区间问题任务法获得的过度自信（错误校准）水平，MICA1 和 MICA2 分别表示通过常识问题与股票市场预测问题获得的过度自信水平。BTA1 和 BTA2 分别表示通过两个不同问题获得的过度自信（自我感觉良好效应）水平。IC1 和 IC2 从两个方面来刻画控制幻觉：对随机事件的控制、对个人成功可能性的过高估计。EO 则表示对未来的过度乐观程度。

Glaser 和 Weber 发现，多种形式的过度自信水平的结果是互相矛盾的。例如，MICA1、MICA2 和 BTA1、IC1 与 EO 之间呈现负相关。因此，在研究的过程中，我们应该尽量细化对过度自信的刻画，避免笼统的概念。

资料来源：GLASET M，WEBER M. *Overconfidence and trading volume*[J]. *Geneva risk and insurance review*，2007，32(1).

6. 与过度自信相关的心理偏差

既然人们已经认识到过度自信普遍存在并且会带来一些额外的成本，那么为什么仍然没有改正错误？人们又为什么会产生过度自信？以下三类行为偏差影响了人们的行为，导致了过度自信的产生。

第一个影响过度自信的行为偏差是自归因偏差（self-attribution bias），即利己主义归因偏差，因为存在归因偏差，成功时人们会认为这是由于自己的能力和努力，而失败时人们则会将责任归因于外部因素。研究证明，这样的归因偏差会增强过度自信程度。一个相同任务的成功和失败，对过度自信程度产生的影响是不对称的。举例来说，假设 A 参加一个演讲比赛，他要取得成功既需要自身优秀的能力也需要评委的认可，而评委即为外部因素。当 A 获得好的名次时，他会认为自己受到奖励是因为自己的努力，而低估外部因素

的作用,假设这样的心理偏差将会导致他的过度自信水平提高 b;而当 A 没有取得好名次时,其过度自信水平会降低 c,但是他会倾向于认为是评委的偏心和比赛的不公导致了这样的结果,从而其过度自信程度的降低要小于前述情况,即 $b>c$。从长期来看,这样不对称影响的积累导致了过度自信。

后视偏差(后见之明,hindsight bias)和确认性偏差(confirmation bias)也是导致过度自信并影响人们纠正的两个重要因素。后视偏差,即事后诸葛亮,是一种记忆歪曲,是指在事情发生后,人们总是会倾向于认为自己事先的判断是准确的,类似于平常所说的"马后炮"。确认性偏差与后视偏差相似,是指人们倾向于去寻找能支持自己事先判断的证据或者在理解信息时对支持自己判断的信息赋予更高的比重,而选择忽视反面证据。这样的信息搜寻策略不仅使得与自己判断一致的信息容易被提取,也将改变人们对模棱两可的信息的理解。除此之外,动机因素更是加强了这样的效应,人们总是认为自己是有智慧的,因此在处理信息时会不自觉地优先处理与自己想法一致的信息。例如,相信星座的人会下意识地去搜寻与星座描述相符的行为而选择性地忽略一些与之相违背的事实。后视偏差和确认性偏差都说明人们倾向于认为自己的判断是正确的,并且这种错觉是下意识的,而这也会进一步增强过度自信。

除此之外,投资者在失去判断标准的时候,也会表现出过度自信,并且由于羊群效应,这种非理性行为之间还会相互模仿,在市场扩散。

7.2 过度自信与投资者行为

在现实生活中,投资者并不是完全理性的,非理性投资者非常显著的一个特征就是过度自信,那么过度自信对投资决策到底有怎样的影响?对金融市场又有怎样的影响呢?

1. 过度自信与投资决策

过度自信会影响人们的投资决策,那么过度自信影响投资决策的机制是怎样的呢?

投资决策的过程可以被分为以下三阶段:判断(judgement)、估值(evaluation)和投资决策(investment decision)。

在判断阶段,投资者形成对金融资产(通常是股票)的最初印象和风险感知,而信息来源可能仅仅是资产的名字这样简单的途径。

在估值阶段,投资者获得有关该金融资产的更多信息,如该公司的财务信息、历史价格信息以及其他重大事件信息等,投资者可能会利用各种估值方法对股票进行估值,并产生对未来股价的预测。

在投资决策阶段,投资者将资金投资于所选股票或者无风险资产。

过度自信和风险态度都会影响投资者的决策,而风险厌恶和过度自信的作用是相反的。图 7.1 展示了一个简单地将过度自信和风险厌恶引入投资决策各个阶段的模型。

过度自信会正向地影响投资决策。过度自信的投资者可能会产生过度自信的判断或者估值,此时过度自信对决策的影响是间接的,值得注意的是,投资者在实际决策时,并不一定会与之前判断或估值的结论一致。当然,过度自信也可以直接在投资决策阶段发挥

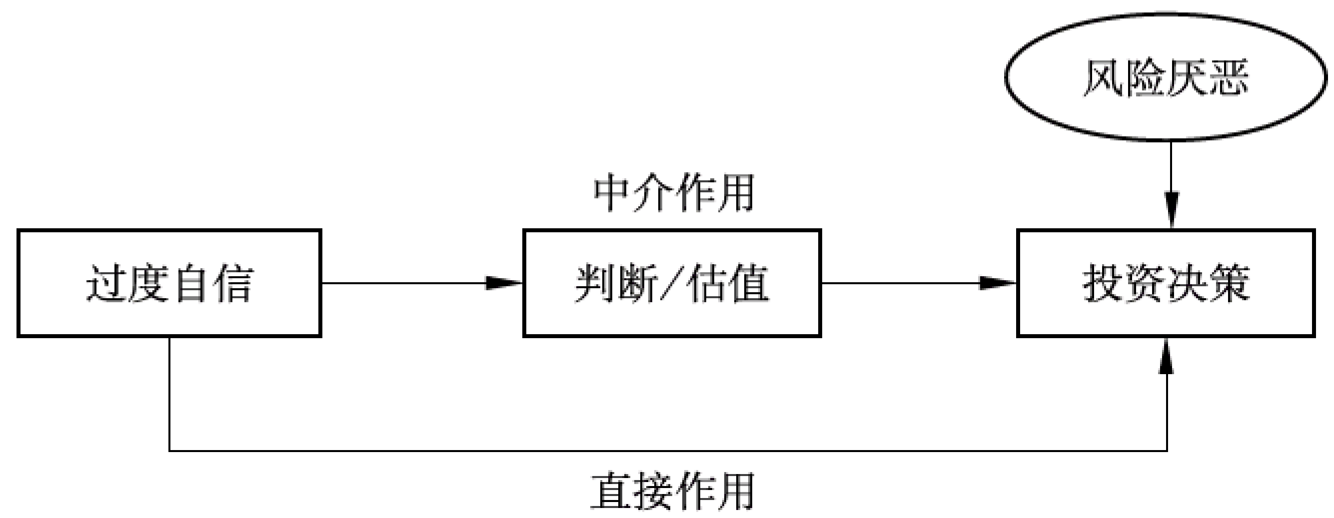


图 7.1　投资者作出投资决策的三个阶段

作用，投资者过度相信自己所掌握的信息，从而持有分散化不足的投资组合或者采用激进的投资策略。除了过度自信以外，风险厌恶也会影响投资决策，它的影响与过度自信相反，较高的风险厌恶水平会直接并且负向地影响投资者对风险资产的投资，高风险厌恶投资者更倾向于将资金投入无风险资产而非风险资产。

Lambert 等应用上面的模型研究了过度自信和专业知识(expertise)在投资决策中起到的作用。他们通过实验来模拟决策的过程，实验被试在判断阶段根据公司名字对 20 只股票给出 5 档得分以表示他们对公司(股票)的熟悉程度和风险接受程度；在估值阶段，他们需要对公司股票进行估值预测，并估计预测的准确性和难度；在投资阶段，实验被试可以将 100 000 欧元在 20 只股票和无风险资产之间进行分配。研究发现，对于普通人和金融专业人士，过度自信起作用的阶段是不同的。对于普通人(以学生为代表)，过度自信主要影响判断阶段，而对最终的投资决策起主导作用的是风险厌恶水平，这是因为估值阶段让学生对自己最初的判断产生怀疑从而进行"二次思考"。对专业人士(以银行从业者为代表)而言，过度自信影响他们的估值和决策阶段，因为专业人士在进行决策(特别是像投资决策这样的重要决策)之前，会习惯性地采用一系列的流程(如估值等)，他们的估值结果和最终决策是有很大的联系的，因此过度自信的中介作用比较明显。

2. 过度自信与过度交易

每一个交易日，资本市场都发生着成千上万笔交易。从理性的角度考虑，资本市场巨大的交易量无法仅从对冲和流动性需要方面来解释，因此学者们试图用一些非理性的因素来解释这样的现象。大量研究表明，过度自信会导致过度交易(excessive trading)，并导致投资组合收益的下降。事实上，过度交易本身就带有贬义，指导致收益率降低得过多的交易。过度自信既会影响证券交易频率，也会影响交易量。过度自信的投资者认为自己具有专业的知识和能力，希望挑选出高于市场平均收益的资产，并且对自己的判断过于自信，从而导致交易频率和交易量的提高。

有学者提出，积极的想法(如膨胀的自尊自信和过度乐观)能够使人们更加努力地工作，从而导致更好的结果。但是这一理论在金融市场却是完全不符合实际的，投资者的过度自信会大大降低他们的投资表现，甚至对于专业人士也是一样。过度自信的交易员与其他交易员相比，赚得更少。金融市场似乎会对过度自信进行惩罚，而不是奖励。

在实证领域对过度自信与证券资产交易的研究相当丰富。较早期的研究通常没有直接对过度自信进行测度，而是通过寻找过度自信的代理变量(如性别、历史交易信息等)来

进行研究。

Barber 和 Odean 在 2000 年的研究中检验了过度交易是否会降低投资组合的表现和投资者的期望效用。通过检验超过 60 000 名个体投资者在 1991—1996 年的历史交易信息,他们发现在考虑了交易成本之后,个体投资者的投资表现要差于相关的市场投资组合,并且交易量越大,收益表现越差,与理论模型的结论一致。在 2001 年的研究中,他们用性别作为过度自信程度的代理变量,检验了"过度自信会导致过度交易"这一假说。心理学研究表明男性比女性更倾向于过度自信,特别是在男性占优势的金融领域;而过度自信的投资者会高估自己信息的准确性从而高估预期的交易收入,他们甚至会在预期净收益为负的情况下进行交易。因此,和女性相比,男性的证券交易量应该更大,并且投资表现更差。他们通过实证研究证明了这一结论。图 7.2 所示的柱形图显示了男女两性在换手率和股票净收益率方面的差异,男性的换手率明显高于女性,而男性的股票净收益也明显低于女性,并且在单身男性和单身女性中,这一差距更加显著。

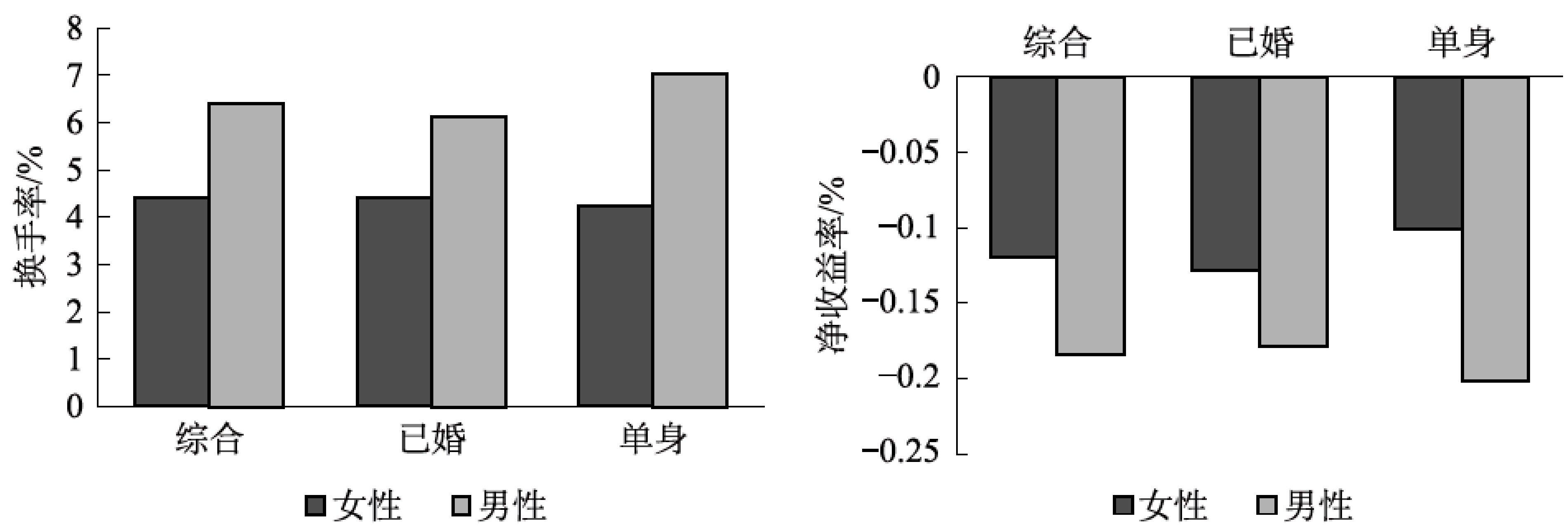


图 7.2 男性、女性在换手率和股票净收益率方面的差异

随着心理学界对过度自信方面的研究逐渐完善,越来越多的研究开始直接检验过度自信与交易行为的关系。这类研究通常会精心设计出能够测度过度自信的心理学方法来获得投资者的心理数据。

Grinblatt 和 Keloharju(2009)做过一个非常经典的研究。他们利用来自芬兰军队的心理测试数据、投资者税务信息和驾驶记录以及投资者的股票交易记录,研究了两种心理特征——感觉诉求(sensation seeking)和过度自信对投资者股票交易的影响。感觉诉求是一种寻求刺激、多样化和高强度的感觉与经历,并愿意为此承担更多心理、社会、法律以及经济上的风险的性格特征。他们认为感觉诉求和过度自信是投资者愿意承担分散化不足策略带来的风险进行频繁交易的推动因素。一方面,对刺激的追求使得投资者从投资组合的变动中获得乐趣;另一方面,投资者过高地估计了自己的投资能力,从而认为自己可以从交易中获得更高的收益。Grinblatt 和 Keloharju 用投资者驾驶记录信息中的超速驾驶罚单数量作为感觉诉求的代理变量,并利用芬兰男性参加兵役时的心理测试数据衡量他们的过度自信程度,当这些数据和股票交易记录匹配起来时,他们发现感觉诉求者和过度自信者有更高的股票交易频率。

在传统的投资模型中,投资者在获取信息上投入得越多,其证券交易就会越频繁。这是因为投资者通过获取信息来使自己变得"知情",那么他一定会希望通过持有更多的风

险资产、获取超额收益来补偿其信息获取成本。信息收集的结果要么是获得了更多的信息,要么是对原有信息的进一步加强,而这些都会促使投资者去更频繁地交易,使得信息获取与证券交易频率间存在正相关的关系。Abreu 和 Mendes(2011)探究了过度自信程度与信息获取渠道对这种关系的影响。他们的研究再一次证明了过度自信(基于自我感觉良好效应)与交易频率之间的正相关关系。并且,虽然投资于信息会促使证券交易更加频繁,但是信息来源对不同投资者的投资行为有着不同的影响:对于非过度自信的投资者,当他们获得来自专业机构的信息时(如专业出版物或者交易所的公告),从理性的角度出发,他们的交易频率会提高;而对于过度自信的投资者,当他们的信息主要来自亲朋好友口耳相传,他们的交易频率会有所降低,可能的原因是他们通过与周围朋友的比较发现自己投资失败并不仅仅是因为运气不好,认识到自己能力的不足,从而降低了过度自信程度。[①]

总的来说,过度自信对投资者在资本市场的交易有很大的影响。过度自信使投资者轻视投资风险,从而采取相比理性投资者而言更加激进的投资策略,导致过高的交易频率和过大的交易量,考虑到交易成本,投资者的投资收益反而因此下降。

专栏 7.2 风险 VS 模糊性:过度自信、性别与金融交易

风险(risk)和模糊性(ambiguity)是两种不同的不确定性(uncertainty)状态,前者已知不同收益结果的分布,后者则未知。我们以浙江大学的学生为实验被试,探究在模糊性和风险这两种不同的情况之下,过度自信对金融市场投资行为的影响。实验的设计是这样的:首先通过问卷测量学生在错误校准、自我感觉良好效应和控制幻觉三方面的过度自信水平,然后请学生参加一个基于双向拍卖(double-auction)的模拟金融交易市场。10 个实验被试组成一个持续 10 轮交易的市场。实验被试可以出价卖出或买入股票,或者他们可以作为价格接受者,接受市场上他人的出价。在每一轮结束后,将会向持有股票的实验被试发放股利。模糊性和风险两种情况的差别在于,在风险场景下,实验被试知道股利的分布情况(0、8、28 和 60 各有 25%的可能性出现),而在模糊性场景下,实验被试只知道股利有四种可能性,并不知道每种可能性出现的概率。

实验结果发现,只有当投资者不知道未来股利的分布,也就是在模糊性情况下,过度自信才对交易量有显著的影响。这是因为当投资者不知道股利未来的分布时,他们不仅需要根据自己的模型来预测资产收益,还需要预测股利的波动性,因此要进行价值估计就更加困难。外部的不确定性会导致过度自信的投资者对自己的模型的依赖性更加严重,因此导致更为严重的过度交易。

此外,我们还发现,在自我感觉良好效应方面,男性比女性体现出更明显的过度自信。在风险场景下,男女交易量没有显著差异,而在模糊性场景下,男性比女性交易量更大。这说明在模糊性场景下,过度自信和性别对交易量的影响是一致的。

资料来源:杨晓兰,朱励.*Ambiguity vs risk:an experimental study of overconfidence,gender and trading activity* [J]. *Journal of behavioral and experimental finance*,2016(9):125-131.

---

① 注:数据样本期间为金融危机之后 6 个月,在这一阶段,大多数投资者都遭受了不同程度的损失。

3. 过度自信投资者的生存问题

在金融市场中，过度自信会导致错误的判断、过度的交易和更差的投资表现，那么为什么过度自信的交易者还能长期地存在于金融市场呢？

首先，过度自信的投资者会低估风险，他们可能会持有更多的风险资产或者风险更高的资产，而高风险的资产会获得更高的期望收益，因此过度自信的投资者就有可能比理性投资者表现得更好。

其次，风险厌恶的过度自信投资者会在已有信息的基础上采取更积极冒险的交易策略。因此，过度自信的交易者更有可能发掘风险盈利机会，这些机会可能是由投资者的流动性需求造成的，也有可能是由于噪声交易者的错误定价造成的。为什么过度自信的投资者更能发掘风险盈利机会？在下面两种情况下，过度自信的投资者反而“误打误撞”地获得了更高的收益：一是在完全竞争市场中，过度自信投资者所掌握的信息是有效且充足的，那么他们低估风险或者高估收益，采取进取型的投资策略就能带来更高的收益。这种增加的收益来自激进的投资策略本身，而不是因为投资者对这种策略的坚持或者对其他投资者的影响。二是假设投资者所掌握的信息是不充足的，但是在非理性投资者主导的市场中，过度自信非理性投资者的表现反而要好于理性投资者。事实上在中国股票市场中，这样的例子很多。以五粮液的认沽权证为例，在其刚上市的一段时间内，其成交价格远高于布莱克—斯克尔斯期权定价模型的定价，甚至一度高于其行权价，日换手率达到140%。显然此时非理性的因素主导了市场，市场中存在极大的资产泡沫，但是过度自信的交易者却恰恰有可能从中获利（即使他们对权证的概念还没有正确的了解，但是根据已有的历史价格信息，只要不是这场“击鼓传花”的最后一个人，他们是可能从中获得较大盈利的）。中国市场不乏资产价格严重偏离其基础价值却仍然保持着涨势的股票（有的股票市盈率甚至高达几百），理性投资者应该看跌这些股票，但过度自信的投资者却可能“误打误撞”地把握这一投资者错误定价所产生的盈利机会。

最后，从市场整体的角度，选择性偏误和幸存者偏差使得金融市场的投资者趋向于过度自信，而以往的成功经历又进一步加强了过度自信。新投资者不断进入市场，使这一过程不断重复，因此金融市场中过度自信的投资者可能永远存在。

7.3 过度自信与公司行为

既然过度自信普遍存在，那么有理由认为公司的管理层也同样存在过度自信的心理偏差。虽然对过度自信和其他心理偏差已经有30多年的研究历史，但是对于管理层的心理偏差研究却在近年来才引起重视。那么管理层的过度自信与投资者的过度自信有什么不同，要如何去衡量？它对公司的决策以及发展又有什么影响呢？

1. 管理层的过度自信

很多心理学的文献都指出公司管理层更容易出现过度自信的特点[①]，具体表现为低估

① 此处管理层的过度自信、高管的过度自信与公司行为金融中经理人的过度自信具有相同的意思。

项目风险，高估项目回报，高估公司价值以及自身能力，主要原因可以归结为以下几点：第一，过度自信的管理者更容易受到提拔并留在公司管理层，而 CEO 等高管通常从公司原有管理层中进行选拔，并且过度自信的管理者更容易被提拔为高管。这是因为过度自信的管理者会选择风险更高的项目，而更高的风险一方面提高了结果的波动性，另一方面也更有可能带来更好的业绩，而那些在风险中获得更好的结果的“胜者”则更有可能受到提拔。因此，从某方面来说，选拔机制是青睐过度自信的管理者的。第二，公司管理层，特别是高管，必然具有一定的成功经历，而因为自归因偏差的存在，以往的成功经历会进一步加剧他们的过度自信程度。第三，公司管理层更容易产生控制幻觉，高估自己的能力而低估失败的可能性。

Graham 和 Harvey 在他们 2001 年发布的 CFO（首席财务官）调查中，直接证明了管理层在决策中存在心理偏差。在华尔街技术泡沫即将被刺破的 1999 年第二季度，仍然有 70%的 CFO 认为他们公司的股票价值是被低估的。在 Ben-David 和 Harvey（2007）的研究中，他们发现仅有 38%的 S&P500 指数收益率落在 CFO 预测的 80%置信区间中，因此他们同样得出 CFO 在错误校准的层面上存在过度自信的结论。

专栏 7.3 管理层过度自信与薪酬合同设计

Gervais 和 Odean 在他们的文章中曾提出过一种模型，他们假设公司本身是知道自己雇用的管理层是否具有过度自信的特征，并且设计不同的薪酬合同来适应管理层的过度自信特征，以最大限度地获取管理层劳动的经济剩余（economic surplus），降低代理成本，即“利用”管理层的过度自信。在他们的模型中，适度的过度自信是有利于公司的。

适度的过度自信带来的好处主要有两个：对于轻微过度自信的管理层来说，公司可以通过让薪酬曲线更加平滑（减少曲线的凸性）来增加公司的价值。和理性管理层相比，较少的绩效薪酬（如红利、股票、期权等）就可以激励过度自信管理层去执行那些能够增加公司价值的风险项目，过度自信使得公司对管理层的激励更加“便宜”。除此以外，过度自信还能够激励管理层付出更多的努力，因为过度自信的管理层高估了自己的努力能给风险项目带来的收益，而这也进一步增加了投资项目成功的可能性。

如果管理层是极度过度自信的，那么相反地，公司就要提供很高的绩效薪酬（薪酬曲线变得非常凸）来雇用这样的管理层，将风险大量转移到管理层身上。

管理层的过度自信程度和公司本身的特质以及薪酬合同是相匹配的。理性与轻微过度自信的管理层更可能在更安全、多元化、高价值的公司中工作，他们的薪酬曲线显得更加平滑；而高度过度自信的管理层则可能被更高的绩效薪酬（convex compensation contract）吸引，从而在那些高风险、业务单一的成长性公司中工作。

资料来源：GERVAIS S，HEATON J B，ODEAN T. *Overconfidence，compensation contracts，and capital budgeting* [J]. *Journal of finance*，2011，66(5)：1735-1777.

那么管理层的过度自信程度应该怎么衡量呢？虽然前面提到的过度自信水平测量方法也可以应用到公司高管身上，但是作为金融研究中一类特殊而重要的群体，人们总是希望能有一些特定的方法来衡量他们的过度自信水平。衡量高管过度自信程度最普遍的方法之一就是从高管对自己持有的公司股票期权的处理决定和持股变化情况出发的。股权

激励已经成为最普遍的激励政策之一，高管们收到的期权通常在执行时间上有所限制(vesting period)，当持有期满足规定时，他们才可以自由处理期权。因为高管所持有的公司股票期权受公司经营风险的影响，因此出于分散风险的考虑，理性的管理者应该及早地执行期权，将预期收益转化为实际收益。而过度自信的管理者则因为高估了公司未来的表现，从而会更愿意持有期权，将自己暴露在公司特定风险下(不同于外部投资者，高管人员无法通过交易自身拥有的公司股权或是卖空股票来对冲风险)。因此，如果高管在限制执行期结束之后，继续持有实值期权而不执行，那么可以认为他是过度自信的。同样地，如果高管人员在一个行权期间内增持股票，那么说明其预期股票价格有较大可能上升，表现出过度自信。但是，由于目前我国上市公司股权激励政策还不完善，很大地限制了这个指标的应用。此外，也有学者用业绩预测偏差程度和高管薪酬集中程度等作为过度自信水平的代理变量。在我国，国家统计局公布的企业景气指数[①]也是一个较为常用的指标。

管理层过度自信的主要指标和代表性文献如表 7.1 所示。

表 7.1 管理层过度自信的主要指标和代表性文献

指标	文献
自愿持有企业股票和期权的倾向	MALMENDIER U,TATE G A. *CEO overconfidence and corporate investment*[J]. *Journal of finance*,2005,60(6):2661-2700
业绩预测偏差程度	LIN Y,HU S,CHEN M. *Managerial optimism and corporate investment: some empirical evidence from Taiwan*[J]. *Pacific-basin finance journal*,2005,13(5):523-546
消费者情绪指数	OLIVER B R. *The impact of management confidence on capital structure*[J]. *Social Science Electronic Publishing. SSPN working paper*,2005(791924)
企业景气指数[①]	余明桂,夏新平,邹振松. 管理者过度自信与企业激进负债行为[J]. 管理世界,2006(8):104-112
高管薪酬集中程度	HAYWARD M L,HAMBRICK D C. *Explaining the premiums paid for large acquisitions: evidence of CEO hubris*[J]. *Administrative science quarterly*,1997,42(1):103-127. 姜付秀,张敏,陆正飞,等. 管理者过度自信、企业扩张与财务困境[J]. 经济研究,2009(1):131-143
主流媒体对管理者的评价	MALMENDIER U,TATE G. *Who makes acquisitions? CEO overconfidence and the market's reaction*[J]. *Journal of financial economics*,2008,89(1):20-43. MALMENDIER U,TATE G A,YAN J. *Overconfidence and early-life experiences: the effect of managerial traits on corporate financial policies*[J]. *Journal of finance*,2011,66(5):1687-1733.
管理者个人特征	余明桂,李文贵,潘红波. 管理者过度自信与企业风险承担[J]. 金融研究,2013(1):149-163

① 企业景气指数是通过对部分企业负责人定期进行问卷调查，并根据他们对企业经营状况及宏观经济环境的判断和预期来编制的指数，从而能够比较直观地反映特定企业管理者对自身企业的预期，是一个主观判断的过程，与过度自信的表现比较一致。

2. 过度自信与公司决策

管理层的过度自信，特别是 CEO 的过度自信，对公司的投资和融资决策都有着很大的影响。这部分内容将会在本书第 12 章行为公司金融章节做详细的介绍。

1）公司投资对现金流的敏感性

在实际中，人们会发现公司投资和自由现金流之间存在正相关关系。当公司内部资金充足时，公司投资会增多；而当公司面临财务限制时，投资会显著减少。根据公司金融理论，只要是净现金流为正的投资，都应该被执行，与这一资金是来自公司内部还是外部无关。现金流的限制使得公司所做的投资决策不是最优的，这实际上是一种投资扭曲的异象。

管理层与股东利益不一致和公司内部与资本市场之间信息不对称是这一投资扭曲现象的两个传统解释：第一，根据代理理论，为了获取自己的私人利益，经理人会进行过度投资。相较于外部融资，内部充足的自由现金流为经理人进行利己的投资提供了更大的方便，因此造成了这一投资扭曲。第二，在信息非对称的情况下，如果公司股票是被低估的，那么经理人为了防止股票被稀释，就会尽量减少外部融资（如增发股票）。这时，自由现金流的增多就会增加公司投资，降低投资扭曲。

除了以上这些理性解释外，管理层过度自信也能够解释企业投资扭曲现象。公司 CEO 和市场对公司价值的看法影响了公司的投资。与代理理论不同的是，行为金融理论认为过度自信的管理层与股东利益是一致的，并且他们愿意自己投资于公司股票。过度自信的 CEO 会系统性地低估投资风险而高估投资项目的回报，因此如果他们有充足的内部资金而不必受到资本市场的约束，他们就会有过度投资的倾向[①]，其中一些投资项目对于投资者来说甚至可能是不利于公司价值增长的。但是，如果他们没有充足的内部资金，他们将不愿意发行新的股票，因为他们认为公司的股票是被市场低估的，因此公司的投资就会受到一定程度的限制。然而究竟公司投资会受到怎样的影响则取决于对投资项目的过度自信和对公司本身的过度自信这两种力量的平衡作用。对于过度自信的 CEO，投资的标准不是客观的净现金流水平而是他们内心对投资和收益的主观判断，因此造成了投资扭曲。

实证研究也证明了管理层过度自信假说的正确性，Malmendier 和 Tate（2005）通过构建多个 CEO 过度自信的代理变量对此进行了验证。他们对 CEO 过度自信水平的测度与表 7.1 提到的基于经理人主动持有可执行期权的意愿的测度类似，他们还考虑了这种行为的持续性以及他们在期权上的盈利和损失。他们对投资水平与现金流、CEO 过度自信水平以及现金流和过度自信水平的交互项进行回归，结果显示投资对现金流的敏感度与管理层过度自信之间存在强烈的正相关，即使采用不同的过度自信测度方法结果也依然稳健。这样的结果不仅验证了管理层过度自信对投资扭曲现象的影响，也给传统的股权和期权激励方式打了一个大大的问号。对于过度自信的管理层来说，这样的激励可能是

① 过度自信的管理层会高估投资项目未来的现金流收入，而低估折现率（项目资金的成本），使一些本来 NPV（净现值）为负的项目可能成为 NPV 为正的项目，从而，导致过度投资于一些本来应该被拒绝的价值毁灭（value-destroying）项目。

无效的，甚至可能进一步加大投资的扭曲。

2）兼并收购

已经知道管理层的过度自信会对公司投资决策产生影响。心理学家指出人们在认为事件结果受自己控制或者为某件事付出很多努力时更容易表现出过度自信，而兼并收购同时符合以上两个标准，接下来将具体地考察过度自信到底是怎样影响兼并收购决策的。

假设CEO过度自信普遍存在，那么在并购中，CEO的过度自信到底发挥了怎样的作用？Malmendier和Tate在2008年的研究试图回答这个问题。过度自信的CEO会高估项目的预期收益和自身的能力，在公司并购中，他们表现出"自我感觉良好"的心理现象，认为他们管理企业的能力比目标公司要强，导致他们产生能控制结果的错觉而过低地估计并购失败的概率。但是他们同时也会认为自身公司是受外界低估的。第一个因素使得管理层更多地参与兼并收购，而第二个因素则使管理层不愿意通过外部融资来参与兼并收购。因此从理论上来说，管理层过度自信对兼并收购的影响是不确定的。

延伸阅读7.1 过度自信与财务误报——滑坡理论

一项美国的研究显示，在49家于1990—2000年违反美国证券交易委员会(SEC)会计与审计起诉公告(AAERs)的公司中，只有25%被定义为诈骗，而大多数(75%)则只被定义为财务误报(misreporting)。事实上，公司的财务误报可能与公司管理层的过度自信有关。

首先，过度自信的管理层会倾向于延迟损失确认并采取相对宽松的财务准则。在初期，过度自信的经理人高估了公司未来的收益，但是此时公司的财务误报通常是比较微小且无意的。但是在接下来的一段时间里，不理想的财务状况(收益实现不佳)则进一步加强了财务误报。一方面，过度自信的经理人依然对公司未来表现保持着过度的乐观，并相信公司未来的收益会越来越接近预期，从而产生了进一步的误报。另一方面，经理人可能意识到了财务误报，但是却不得不有意地通过加大误报的方式来掩盖误报的事实。每一期的过度乐观都会加强财务误报的程度，最终误报扩大至足以引起美国证券交易委员会的注意。这就是滑坡理论(slippery slope)。

事实上，不仅是在美国，即使在中国，管理层过度自信也与会计政策的稳健性有着很大的相关性。研究发现，过度自信的管理层会采取稳健性较低的会计政策，逻辑与滑坡理论相似。一方面，过度自信的管理层会高估项目价值，从而产生过度投资，甚至投资于一些净现值为负的项目，而管理层为了掩饰自己的错误，可能会采取提前确认收入、延迟确认损失、低估损失的会计政策。另一方面，过度自信的管理层也可能会低估融资风险而选择激进的融资方式，让公司面临更大的财务困境风险，加大未来融资难度和成本，从而迫使管理层采取不稳健的会计政策。由于国有企业存在所有者缺位和债务软约束的情况，国有企业提供稳健性会计信息的动力不足，管理层过度自信对会计稳健性的影响比民营企业更强。

资料来源：SCHRAND C M，ZECHMAN S L. *Executive overconfidence and the slippery slope to financial misreporting*[J]. *Journal of accounting and economics*，2012，53(1)：311-329.

Malmendier 和 Tate 采用经理人持有可执行公司期权的意愿来作为过度自信的代理变量。研究发现，过度自信的 CEO 进行并购的可能性要比非过度自信的 CEO 高 65%，并且当公司内部资金充裕时（不需要外部融资），这种效果更加显著。两种因素对并购频率的影响与市场对公司的估值有关，当公司的市场估值较高时，第一个因素，即低估并购失败概率起主要作用，即使需要外部融资，管理层也愿意参与并购。他们发现市场对过度自信经理人参与的兼并收购的反应也更加激烈。当宣布兼并收购公告时，过度自信经理人所在公司的股价下跌 90 个基点，而对非过度自信的经理人，市场的反应仅为股价下降 12 个基点。

经理人过度自信对兼并收购的影响主要有：第一，增加兼并收购频率，特别是当公司内部资金充裕时，过度自信的作用更加明显；第二，因为过度自信的经理人高估了兼并收购带来的协同效应，他们更有可能执行不利于公司价值的兼并收购，并且高估目标公司的残留价值，当目标公司有较强的议价能力或者出现了其他竞争对手时，他们可能会出价过高；第三，当公司财务紧张时，过度自信对投资的敏感性影响更大，管理层可能因为不愿意借助外部融资而错过好的兼并收购机会；第四，公司决策者在并购活动中可能低估企业文化冲突的破坏作用。

延伸阅读 7.2　管理层的过度自信与创新

我们可以发现大多数的研究都表明管理层过度自信会损伤公司价值，但是为什么过度自信的管理层会受到提拔并且广泛地存在，为什么高增长的公司更喜欢聘请过度自信的 CEO？过度自信的管理者会更多地参与高风险的投资项目，在一定程度上可以提高投资收益，增加股东权益。在 Hirshleifer 等（2012）的研究中，他们给出了另一个可能的解释：过度自信的 CEO 有利于公司创新。

过度自信的 CEO 对高风险、有挑战性的项目更有热情，而企业的创新项目，不管是创新商业模式还是开发新的技术和新的产品或服务，都是符合这些特征的。Hirshleifer 等研究者通过研究 1993 年到 2003 年 CEO 过度自信与公司创新的关系，发现过度自信 CEO 所在的公司在创新方面投入更多（以 R&D 费用衡量），获得了更多的创新产出（以专利数衡量），其股票的收益波动性也更大。在考虑到管理层过多地投入研发，获取专利可能并不能转化为实际生产力以使公司价值增加，他们控制研发投入不变，结果发现过度自信管理者所在的公司仍然在专利数方面表现出优势，即管理层过度自信有利于提高创新效率。他们认为这是因为理性的风险厌恶管理者可能会错过一些高风险的项目，而过度自信能够鼓励适当的风险承担。并且过度自信的管理者更能够发掘公司潜在的成长机会，并将其转化为公司价值。但是，这样的积极作用只发生在创新性行业。这是因为过度自信的作用具有双面性，它既能够促进企业创新、增加公司价值，也会导致过度投资从而损害公司价值。创新性行业的公司内部有着较强的发展动力和较多创新机会，因此管理层过度自信对公司价值的积极作用能够凸显。而对于缺乏发展机会的传统行业，它的消极作用就更加显著。

国内也有学者研究过类似的问题。过度自信管理者所在企业的风险承担水平更高，

具体表现为三个方面：第一，过度自信的管理者在选择投资项目时表现出一种“困难效应”，他们对自己的才能过度自信，认为只有复杂而有挑战的项目才能和他们的才能相配。第二，过度自信的管理者对投资环境过度乐观，高估成功的概率，低估项目成本，他们更愿意投资于风险项目，因为他们并不认为这些项目风险过大。第三，过度自信能够缓解代理问题，过度自信激励管理者承担更多风险，使他们表现得更为积极主动，这和股东的利益实际上是一致的。而创新活动总是和风险相联系，更有学者提出“风险承担是经济持续增长的根本动力之一”。

有学者发现过度自信的管理者能够减少不确定性风险对R&D投资的负向影响。创新决策的制定和实施，不仅需要管理者本身有较高的能力，也需要他们更具有冒险精神。在高不确定性的情况下，管理者的过度自信能够增强研发对于企业价值的提升作用，他们对于高风险的偏好缓解了企业创新投入不足的情况，从而带来企业价值的提升；而对于处于低不确定性情况的企业，这种效果并不明显，这些企业所处的环境较为稳定，对创新的需求没有那么强，过度自信管理者的非理性特征反而可能在一定程度上损害企业价值。

资料来源：HIRSHLEIFER D A，LOW A，TEOH S H. *Are overconfident CEOs better innovators*[J]. *Journal of finance*，2012，67(4)：1457-1498.

余明桂，李文贵，潘红波. 管理者过度自信与企业风险承担[J]. 金融研究，2013(1)：149-163.

ACEMOGLU D，ZILIBOTTI F. *Was prometheus unbound by chance? Risk，diversification，and growth*[J]. *Journal of political economy*，1997，105(4)：709-751.

课程实验指引

过度自信水平主要通过心理实验来测试获得。对于有兴趣进一步在实验方面了解过度自信的同学，可了解以下实验设计指引。

1. 实验方法确定

过度自信因为其表现方式的不同，有不同的测度方式，其中最常用的是以错误校准为标准测度的过度自信程度。在一般的研究中，过度自信是作为影响投资者行为的一种因素引入研究的。例如，在研究过度自信与交易量的关系、过度自信对收益率的影响中，需要通过实验来测度过度自信水平。根据理论模型中对过度自信的定义，应该选择与其定义最接近的测度方法，以使得实验数据能够真正反映理论模型。当然，在无法确定最优测度方法，或者希望检验不同测度方法下的过度自信对结果的影响(如稳健性检验)的时候，可以通过实验获得不同的过度自信指标(如错误校准、控制幻觉、自我感觉良好等)。

2. 实验设计

实验通过问卷来获取个体的相关心理状态，并通过相关的计算进一步确定过度自信的指标值。通常衡量过度自信的问卷不会作为一个单独的研究内容出现，它可能与股票模拟交易实验一起出现。问卷的内容设计与问卷的对象密切相关，问卷内容要根据不同的对象进行相应的修改。

1）错误校准

错误校准主要是指人们对于自己的知识能力以及私人信息的准确性给予过高评价。

这里将采用置信区间问题任务法对实验被试的错误校准程度进行测量，一般实验中会设置10～20道题目。问卷中，实验被试被要求就一些常识性的问题给出90%的置信区间。90%置信区间的含义是他认为正确的值有90%的可能性落入他所给出的区间，若正确答案包含于该区间内，则认为实验被试回答正确。90%只是一个比较常用的置信度，也可以根据自己研究的需要而改变置信度。

整个置信区间问题任务共有20道问题。通过统计20道题的正确率，可以得出实验被试错误校准的程度。实验被试的错误校准程度为1减去实验被试的正确率（如若实验被试回答对了8道题，其正确率为0.4，则错误校准程度为0.6）。根据90%的置信区间可知，如果实验被试的错误校准程度为0.1，则他的自信水平是适当的，既没有表现出过度自信，也没有表现出自信不足。数值越大，说明实验被试的错误校准程度越高。问卷中的20道题目如表7.2所示。

表7.2　置信区间问题任务法题目参考

题目	区间下界	区间上界
1. 2010年的世界人口相比2000年增长了多少/%		
2. 牛顿发现万有引力的年份		
3. 东方明珠塔的高度/米		
4. 联合国的国家数量		
5. 太阳的寿命/亿年		
6. 人体共有多少块肌肉		
7. 2012年中国人口的老龄化率/%		
8. 世界陆地面积约为多少万平方公里		
9. 中国的南北跨度是多少公里		
10. 亚洲象的平均寿命/年		
11. 莫扎特第一首交响乐诞生的年份		
12. 欧盟的成员国数量		
13. 希腊在1896年本土举办的第一届夏季奥运会中获得的金牌数		
14. 2012年全球森林覆盖率/%		
15. 古龙的小说数量		
16. 2010年回族的人口为多少万人		
17. 大西洋的最深处为多少米		
18. 马拉松世界纪录为多少分钟		
19. 爱因斯坦提出狭义相对论的年份		
20. 北约的成员国数量		

2）控制幻觉

控制幻觉主要是指过高预期了利于自身情况发生的可能性，从而产生一种能够对周围事物发生控制的错觉。这里对控制幻觉程度的度量将采用两道问题。实验被试被要求用1～5表示对以下两个问题的同意程度（1～5分别表示从完全不同意到完全同意）。设

置的两道题目如下。

(1) 我买的股票将来不会表现不佳。

(2) 我不知道哪些股票未来的表现会在平均水平之上。

实验被试的过高估计程度为问题(1)的得分减去问题(2)的得分,数值越大,表明过高估计程度越高,数值为0时表示实验被试具有合适的自信水平。

除了上述的方式以外,也可以直接取各问题得分的平均值作为实验被试的过高估计程度,但是相应地,设置的问题要进行一定的改变。以下是一些可选的题目:

(1) 我的估值是完全正确的。

(2) 对我来说专注于自己的目标并且实现目标非常容易。

(3) 不管发生什么事,我都可以解决。

3) 自我感觉良好效应

自我感觉良好效应主要涉及自己与他人的对比,过度自信的人会认为自己比平均水平更加聪明或者优秀。首先介绍与股票模拟实验相结合的衡量方法,这里采用直接询问实验被试对自己在模拟股票交易实验中的排名预期的方法来衡量他们自我感觉良好效应的程度。

在这次实验中,实验被试将参与一个由5人组成的模拟股票市场,经过10轮交易后,会产生一个根据初始现金与最终现金计算得出的收益率。在模拟股票交易实验之后,通过以下问题获得实验被试自我感觉良好效应的程度:

你认为会有几个人的收益率高于你(请在0~4之间进行选择)。

实验被试过度自信程度为4减去实验被试对于这道问题的回答。假如实验被试觉得自己的表现是在平均水平之上的,则他的回答将是0~2,其过度自信程度将为2~4;假如实验被试认为自己的表现是在平均水平之下的,则他的回答将是3~4,其过度自信程度将为0~1,表现为自信不足。因此数值越大,则实验被试的过度自信程度越高。

当然除了上述的问题以外,也可以有其他形式的问题,但最关键的是一定要涉及与他人的对比。例如,Glaser 和 Weber(2004)在他们的研究中设计的衡量自我感觉良好效应的问题。

(1) 你认为在你所在的经纪公司中,有多少顾客比你拥有更好的发现优质股票的技能?(给出百分比)

(2) 在你所在的经纪公司中,有多少顾客在过去4年间的股票收益高于你?(给出百分比)

实验被试的过度自信水平定义为$\frac{50-\text{answer}}{50}$,取两题得分的平均数作为最终的得分。若最终得分为正值,则表明实验被试存在过度自信,数值越大,过度自信程度越高;若最终得分为负值,则实验被试存在自信不足。

本章小结

(1) 过度自信是指对自己的知识和能力有过高的估计,过度重视自己所拥有的信息而

忽略市场上或者他人的信息，或者认为自己能够掌控未来的事件。

(2) 过度自信存在多种表现形式，主要包括错误校准、自我感觉良好效应、控制幻觉与过度乐观。

(3) 自归因偏差、后视偏差和确认性偏差是过度自信产生的原因。

(4) 过度自信会影响投资者的决策过程，并对判断、估值和决策阶段产生不同的影响。

(5) 过度自信会导致过度交易，并导致投资组合收益的下降。

(6) 过度自信也会导致投资组合的分散化不足和处置效应等问题。

(7) 公司管理层也会表现出过度自信的特点，可以用管理层愿意持有可执行实值公司期权的倾向来衡量管理层的过度自信程度。

(8) 过度自信的管理层可能会导致过度投资以及过多地参与兼并收购活动。

思考题

1. 明确下列概念的意义并进行区别：

过度自信、错误校准、自我感觉良好效应、控制幻觉与过度乐观。

2. 请解释为什么过度自信会导致过度交易。

3. 为什么过度自信的投资者能够在金融市场中长久地存在?

4. 请举例说明管理层过度自信对公司投资决策会产生什么影响?

第8章

过度反应

引导案例：Facebook数据泄露丑闻引发的过度反应[①]

2018年3月16日，Facebook公司被曝有超过5 000万名用户（接近Facebook美国活跃用户总数的1/3，美国选民人数的1/4）的资料曾在2016年总统大选期间遭剑桥分析公司（Cambridge Analytica）非法用来发送政治广告，对用户进行定向宣传。

Facebook数据泄露丑闻被曝光后的第一个交易日，即2018年3月19日，Facebook股价大跌6.77%，市值缩水360多亿美元；3月20日，Facebook股价继续下跌2.56%；3月22日，Facebook创始人马克·扎克伯格发表声明，承认平台曾犯下的错误。Facebook的股价继续下跌，市值在一周之内缩水600多亿美元，如图8.1所示。

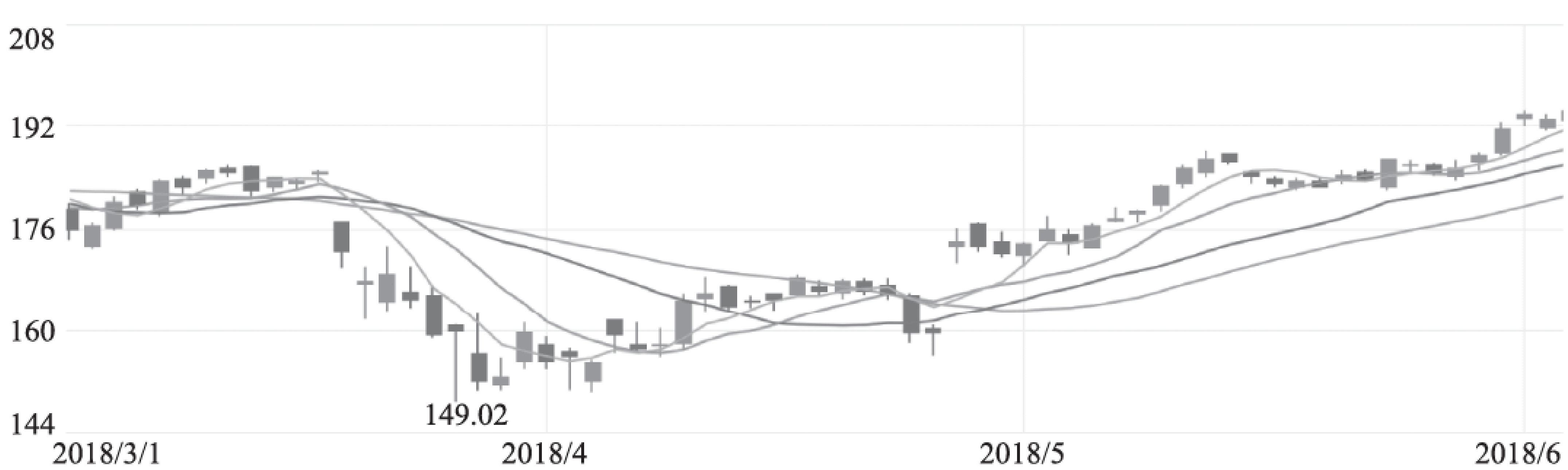


图8.1　Facebook股价日K线图

资料来源：Facebook的股价信息来自新浪财经网。

① 本部分主要参考新浪科技2018年3月22日的新闻《华尔街：股市对Facebook数据泄露丑闻反应过度》。

股票市场对Facebook数据泄露事件的反应是否过度了呢？数据泄露事件到底会对Facebook公司的前景产生多大的影响？事实上，许多分析师认为数据泄露事件不会对Facebook产生太大影响。因为Facebook的用户规模多达21亿，而且很多用户已经对Facebook产生依赖，即使有丑闻曝出，Facebook流失的用户也只是少数，如此庞大的用户规模依然会吸引广告主，Facebook的广告收入不会减少。

许多投资人士认为，股票市场对Facebook数据泄露事件过度反应。Altimeter资本管理公司创始人兼CEO布拉德·格斯特纳认为Facebook CEO马克·扎克伯格和雪莉·桑德伯格是“全球科技行业最优秀的管理团队之一”，建议买入该公司的股票。著名投资人、Legg Mason资产管理公司基金经理比尔·米勒也认为Facebook的股价“太便宜”。加拿大皇家银行资本市场高级科技分析师马克·马哈尼也认同这种看法，他认为此事“基本不用担心”。他还表示，关于监管的担忧已经基本体现在股价中，而广告主可能在未来一年增加在Facebook上的开支。[①]

Facebook的股价在2018年4月之后一路走高，这也表明股票市场对于数据泄露丑闻确实过度反应了。在资本市场，过度反应现象非常常见，投资者对于新信息的反应很难完全理性，容易受到情绪的影响。本章将从过度反应的界定、过度反应的心理学解释、过度反应的理论模型、实证检验和交易制度对反应过度的影响这些角度对过度反应现象进行详细的介绍。

8.1 过度反应的界定

过度反应(overreaction)，或者说反应过度，是指某一事件引起证券价格剧烈变动，超过预期的理论水平，然后再以反向修正的形式回归到其应有价位上的现象。投资者面对突然的或未预期到的事件时，倾向于过度重视眼前的信息并轻视以往的信息，从而引起股价的超涨或超跌，一段时间后投资者能够合理评价、修正事件影响时，股价便会反向修正。

过度反应是证券市场的异象之一，是行为金融领域的重要命题。根据Fama的有效市场假说，在一个有效率的市场中，证券的价格充分反映了所有可获得的信息，证券的价格等于其内在价值，即预期未来现金流的折现价值。De Bondt和塞勒(1985)提出的过度反应假说对有效市场理论构成了严峻挑战。De Bondt和塞勒对美国股市长期回报率的过度反应问题进行了系统的研究，他们将1926年至1982年所有在纽约股票市场上市的股票根据过去3年的累计超常收益率进行排序，将过去3年表现最好的35只股票与表现最差的35只股票分别形成两个投资组合，称为“赢家组合”与“输家组合”，然后考察这两个组合在随后3年中的表现。研究结果表明，在样本期间内，输家组合的超常收益率比赢家组合平均每年高约8%，3年累计高达25%。德博特和塞勒认为过度反应是相对于合适的反应而言的，在此合适的反应是指理性的投资者在对信息一致无偏的基础上使用贝叶斯原则所

① 本部分主要参考新浪科技2018年3月22日的新闻《华尔街：股市对Facebook数据泄露丑闻反应过度》。

作出的反应和行为。过度反应的产生在于投资者并没有使用贝叶斯原则进行客观的信念和行为调整，而是在忽视历史基率的情况下高估新信息的重要性，导致市场价格与基本价值产生过高或过低的偏离。

继 De Bondt 和塞勒的研究之后，有关过度反应的研究越来越多。随着研究的深入，学者发现市场不仅存在过度反应现象，也存在反应不足(underreaction)现象。Jegadeesh 和 Titman(1993)研究发现美国股票市场上单个股票的收益在 3～12 个月的期间内，表现出趋势的连续性，即购买过去几个月中表现良好的证券，同时卖出表现糟糕的证券可以获取超常收益。Barberis，Shleifer 和 Vishny(1998)从价格在利好消息和利空消息到达后的反应趋势出发，对过度反应和反应不足进行了界定。

设在每一时期 t 均有消息 Z 到达，其为

$$Z_t = G(\text{好消息})\text{或 } Z_t = B(\text{坏消息})$$

反应不足是指除最初的反应后，股价的平均收益在利好消息到达的一段时间内比利空消息到达的同样时间内要高的现象，即

$$E(r_{t+1} \mid Z_t = G) > E(r_{t+1} \mid Z_t = B)$$

根据模型假定，新的消息是在 t 期到达的，$t+1$ 期并没有新的消息，因此如果市场是有效的，无论 t 期有好消息还是坏消息，股价在 t 期会完全反应该消息，而在 $t+1$ 期应该没有任何反应，即 $E(r_{t+1} \mid Z_t = G) = E(r_{t+1} \mid Z_t = B)$。然而，$E(r_{t+1} \mid Z_t = G) > E(r_{t+1} \mid Z_t = B)$表明在好消息到达后的第二期，股价继续有正向反应，收益大于坏消息到达后的第二期。这意味着在 t 期股价对新消息的反应不足，在随后会进行部分的修正，价格在当期并没有立即充分地调整到位，消息是逐步反映到价格中的。

过度反应则是指随着利好消息的到达，在 $t+1$ 期股价的平均收益低于利空消息到达后的平均收益的现象，即

$$E(r_{t+1} \mid Z_t = G) < E(r_{t+1} \mid Z_t = B)$$

这表明当利好消息公布后，投资者将会对未来的利好消息保持更高的预期，从而导致股价远远地偏离基本面，形成过度反应。这种过度反应在随后出现反转和修正，所以在好消息到达后的 $t+1$ 期，股价的收益低于坏消息到来后的第二期。

一般来讲，如果股价在变化趋势上存在正的收益自相关性，表明价格对于新信息存在反应不足，而如果股价在变化趋势上存在负自相关性，表明价格对于新信息存在过度反应(后期价格是对前期价格的修正)。

Daniel，Hirshleifer 和 Subrahmanyam(1998)进一步地指出，股价短期存在正的自相关性可能是持续过度反应的结果，从而短期的正自相关性与长期的负自相关性相一致。价格的过度反应和反应不足是相对于均衡价格而言的。当消息到达后，如果价格变化至超过均衡价格，随后向均衡价格方向进行修正，则表明价格对消息反应过头，存在过度反应现象。而当消息到达后，如果价格没有到达均衡价格，随后继续向均衡价格方向靠近，则表明价格对消息反应缓慢，存在反应不足的现象。如图 8.2 所示。

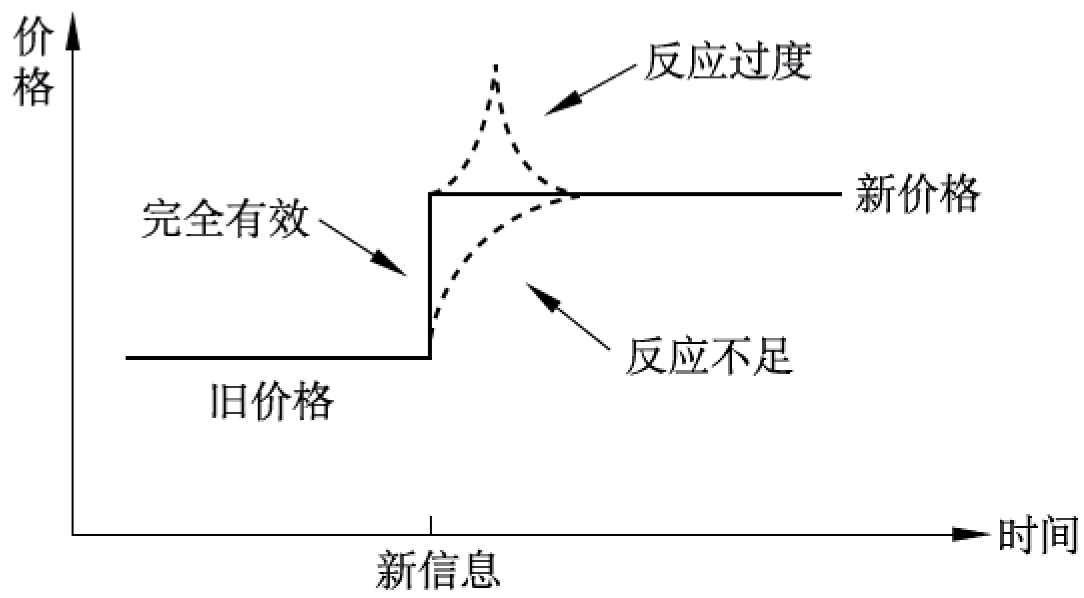


图 8.2　市场对信息的各种反应

专栏 8.1　动量策略和反转策略

学术研究发现股票价格具有短期的动量效应和长期的反转效应，即在短期内股票价格具有延续原来的运动方向的趋势，而在长期内具有发生反转的趋势。行为金融理论把动量效应归因于反应不足，把反转效应归因于反应过度。动量效应和反转效应在股票市场比较常见，投资者利用动量效应构造动量策略，通过在短期内买入过去收益率高的股票，卖出过去收益率低的股票获利；投资者利用反转效应构造反转策略，通过在长期内买入过去收益率低的股票，卖出过去收益率高的股票获利。

动量策略和反转策略的一般步骤：①选择一个时间长度作为业绩评价期，即投资组合的形成期。②先计算形成期各样本证券的收益率，并根据收益率大小进行排序，然后分成若干组，其中收益率最大的一组称为赢家组合，收益率最小的一组称为输家组合。③形成期之后或间隔一段时间后，再选一个时间长度，作为投资组合的持有期。动量策略是买入赢家组合同时卖空输家组合，反转策略是买入输家组合同时卖空赢家组合。④不断重复上述步骤，最后计算动量策略/反转策略各持有期的回报率均值及 t 统计值，如果 t 值表明投资策略的回报率显著大于 0，则投资策略成功。

8.2 过度反应的心理学解释

目前已有一些学者从心理学的角度考察过度反应，研究表明，较多已被证实存在的心理偏差都与过度反应或反应不足具有一定的相关性。

1. *启发式偏差*

德博特和塞勒等学者从行为金融的角度对过度反应现象作出解释，其理论基础是特沃斯基和卡尼曼提出的启发式偏差。特沃斯基和卡尼曼(1974)从实验心理学的角度考察人们在风险环境下的决策行为，认为人们在不确定条件下会采用启发式的思维方法。启发式就是根据以往经验对当前情况作出判断，启发式有时能帮助人们快速地作出准确的判断，但有时会导致判断的偏差。启发式偏差主要有三种：代表性偏差，可得性偏差，锚定和调整。

代表性偏差是指投资者并非按照贝叶斯规则对新信息作出合理反应，而是倾向于对

最近的信息赋予更多的权重,而对整体的基率(base rate)数据赋予较低的权重,这样预测结果将出现很大的偏差,从而导致投资者对最近信息的过度反应。

可得性偏差是指投资者在进行概率估计时,倾向于认为容易想起的事件比不容易想起的事件更常见(容易想起的事件的可得性高),并由此推断可得性高的事件的发生概率大于可得性低的事件。例如,人们对于常见的热门股票可能更易产生过度反应。

锚定是指当人们需要对某个事件做定量估测时,会将某些特定数值作为起始值,起始值像锚一样制约着估测值。在做决策的时候,人们会不自觉地给予最初获得的信息过多的重视。如果投资者进行进一步评价时的判断标准受到当前观测值的影响而被固定,他们对新信息作出的调整往往并不充分,从而可能表现出对消息的反应不足。

2. 保守主义与损失厌恶

Barberis,Shleifer 和 Vishny(1998)认为,保守主义是引起反应不足的重要原因。保守主义是指相对于过去的信息而言,投资者具有低估新信息的倾向,从而不能及时根据情况的变化修正投资预测,这容易导致反应不足。

另外,投资者反应不足的心理学解释还可能是损失厌恶。在损失厌恶偏差的影响下,投资者对损失的敏感程度高于获利,从而趋向于长期地持有亏损的投资,同时又过快地卖出获利的投资。因而当利空消息到达时,价格可能在投资者损失厌恶的影响下表现出反应不足。

3. 过度自信与自归因偏差

Daniel,Hirshleifer 和 Subrahmanyam(1998)认为过度自信和自归因偏差是导致股价短期动量和长期反转的原因。投资者的过度自信,表现为对自身经验预测能力和知识的过高估计。投资者对私有信息的过度自信导致价格的过度反应。

同时,投资者受到自归因偏差的影响,往往倾向于将有利的结果归于自己的能力,进一步助长自信情绪,而将不利的结果归因于外部因素,并不影响其信心水平,从而可能导致价格形成连续的过度反应。

4. 有限理性

Hong 和 Stein(1999)从有限理性的角度解释了过度反应和反应不足现象。他们将市场参与者分为两类:一类是“消息观察者”(new watchers),只根据私人信息判断未来,不依据过去的价格;另一类是“动量交易者”(momentum traders),只根据最近的价格判断趋势。私人信息是逐渐传播开的,因此“消息观察者”的价格反应也是逐渐的、反应不足的,形成了动量效应。这时“动量交易者”参与进来,加深了动量效应,这也孕育着过度反应的发生。

有限理性的投资者不具备处理全部信息的能力,只能处理信息的不同子集。处理不同信息子集的投资者的相互作用,可能导致价格的过度反应和反应不足。

5. 后悔心理与信念坚定

投资者在作出选择时往往会考虑到该选择可能会带来的后悔程度,投资者(包括基金经理)会倾向于买入或卖出别人都在买入或卖出的股票,从而在某些股票上形成过度

反应。

而投资者一旦形成某种观念后，就会在较长时间内坚守这种观念。在信念坚定偏差的影响下，投资者不愿意接触与观念相反的信息，而且一旦投资者接触到该类信息，将采取忽视和怀疑的态度。因此，对新的信息投资者可能反应不足。

由此可见，投资者在现实中受到多种偏差因素的影响，但投资者究竟受到何种因素的影响以及多大程度的影响，与投资者个体不同的特质及其所处的不同市场环境密切相关。

专栏 8.2　坏消息比好消息更让人过度反应

卡尼曼等提出，相比于好消息，人们对坏消息更容易过度反应，恐慌心理使得人们抛售股票，而在熊市阶段，人们受到悲观心态的影响对好消息反应不足。

一些实证研究证实了卡尼曼的观点，但也有一些研究观点与卡尼曼的观点不同。何兴强和李涛(2007)对中国股票市场在牛市与熊市的不同反应进行了研究，他们发现中国股市在牛、熊市阶段对坏消息都反应过度，牛市阶段对好消息反应不足，熊市阶段对好消息比对坏消息的过度反应更明显。

这样的研究结论与中国股市的特点有关，也与研究的时间跨度有关，究竟坏消息是否比好消息更让人过度反应目前还没有定论，但可以确定的是，区分市场对好消息和坏消息的不同反应很有研究意义。

8.3　过度反应的理论模型

针对众多实证检验所揭示的过度反应和反应不足现象，从 20 世纪 90 年代起，行为金融研究者尝试发展理论模型，对过度反应和反应不足现象进行解释。迄今为止，其中最有影响力的模型是 BSV 模型、DHS 模型和 HS 模型。

1. BSV 模型

BSV 模型由 Barberis，Shleifer 和 Vishny(1998)提出，BSV 模型将两种决策偏差纳入投资者的投资行为中，一种是保守主义(conservation)，即相对于过去的信息而言，投资者具有低估新信息的倾向，从而投资者不能及时根据情况的变化修正投资预测，导致反应不足。另一种是代表性偏差(representative)，即投资者按照小数法则行事，以小样本的性质来判断总体样本的性质，从而过分重视近期数据的变化而忽视总体历史数据情况，导致过度反应。

在 BSV 模型中，假定公司的盈利状况是遵循随机游走规律的，同时模型中的投资者并没有意识到公司的盈利状况是遵循随机游走规律的。投资者趋向于认为公司盈利反复运动在两个区域(regimes)或状态(states)中，在不同的状态下，投资者依据不同的模型对公司盈利进行观察和预测。

投资者认为盈利由以下两种模型决定。

模型Ⅰ：公司盈利围绕一定的均值变化。

模型Ⅱ：公司盈利表现出趋势性的变化。

模型Ⅰ和模型Ⅱ都不是随机游走模型，模型Ⅰ和模型Ⅱ的区别在于其中的盈利变化的可能性不同。在模型Ⅰ中，投资者认为公司盈利的冲击在下一期会朝向相反的方向，即本期盈利的正向冲击极有可能在下期不能持续，从而在下期产生负向冲击。在模型Ⅱ中，投资者认为公司盈利的冲击在下一期仍会继续朝向相同的方向，即本期盈利的正向冲击极有可能在下期继续产生连续的正向冲击。

使用模型Ⅰ进行公司盈利预测的投资者对于公司的盈利公告消息反应强度较低，在某种程度上表现出保守主义的倾向，出现反应不足的现象。使用模型Ⅱ进行预测的投资者由于认为公司盈利将在下期继续增加，对于公司的盈利公告反应强烈，其表现类似于受到代表性启发式偏差的影响，出现过度反应的现象。

与此同时，投资者认为公司盈利的模式将会在两种模型间进行转换，尽管这种转换的概率较小。无论投资者是使用模型Ⅰ还是使用模型Ⅱ来对待本期的公司盈利消息，投资者会利用观察到的公司盈利的连续变化来更新其使用的模型。例如，使用模型Ⅰ的投资者在观察到本期公司盈利的变化与上期相反时，会继续认为使用模型Ⅰ是正确的方式，但当其观察到公司盈利出现连续的两个同方向的冲击后，其可能会逐渐倾向于认为公司盈利变化适合模型Ⅱ，并适时对模型进行更新或转换使用模型Ⅱ。BSV模型引入一些参数来分析投资者在两种状态间的转换概率，并根据转换概率考察了反应不足和过度反应出现的情况。

BSV模型可以用来解释股价在短期内的惯性效应和长期的反转效应。短期：受保守主义影响，表现为反应不足，如对盈利的正向冲击反应不足，股价产生惯性效应。长期：经历一系列的正盈利冲击，投资者会调整自己的保守特征，利用代表性法则推断盈利有增长趋势，不断推高股价，最终股价产生反转。

2. DHS模型

Daniel，Hirshleifer 和 Subrahmanyam(1998)从信息的角度建立和发展了描述投资者行为的模型，简称DHS模型。DHS模型着重考虑投资者在阐释私有信息而不是公开信息时存在的偏差，该偏差主要有过度自信和自归因偏差。

DHS模型认为在过度自信的影响下，投资者会对私有信息产生过度自信，从而低估私有信息存在的错误性。投资者根据私有信息进行决策和操作，导致股价被推至远离基本面的价格，此后随着包含噪声的公开信息的到来，股价缓慢进行回归，从而产生股价的长期反转现象。在有偏的自归因作用下，当公开信息或事件符合投资者的私有信息时，投资者倾向于将其归结于自身的能力，并导致投资者的自信程度加强；反之，投资者对不符合私有信息的公开信息和事件并不在意，而是将其归结为外部噪声，并在一定程度上继续保持对私有信息的自信，从而导致短期的动量和长期的反转。

DHS模型的阐述主要分为两部分：信心不变时的基本模型和信心受结果影响的情况。

1）信心不变时的基本模型

在基本模型中，将投资者划分为两种状态：一种是获得信息的有识(informed)投资者，设为I，另一种为没有获得信息的无识(uninformed)投资者，设为U。同时假定有识投资者是风险中性的，而无识投资者是风险厌恶的。

存在四个时期，在时期 0，投资者具有相同的禀赋和一致的先验信念，并各自出于最优化风险转移的目的进行交易。在时期 1，I 获得一个有关标的股票的含噪声的私人信息，并以此与 U 进行交易。在时期 2 和时期 3，一个含噪声的公开信息到达，由此引发了进一步的交易。在时期 4，结论性的公开信息到达，该证券支付了清偿股利。

图 8.3 描述了 DHS 模型的主要内容。粗实线代表信心不变时的价格路径，粗虚线代表有偏自归因影响时的价格路径，细实线代表完全理性预期时的价格路径。在时间 1，知情者得到一个有噪声的私人信息，过度自信导致他对这个信号过度反应，股价高于合理水平。在时间 2，有噪声的公开信息到来，部分地修正偏离的价格，时间 3 另一个有噪声的公开信号到来。在时间 4，确定的公开信息到来，价格完全被修正。图 8.3 的峰和谷之前所对应的部分称为“过度反应阶段”，之后的部分称为“修正阶段”。

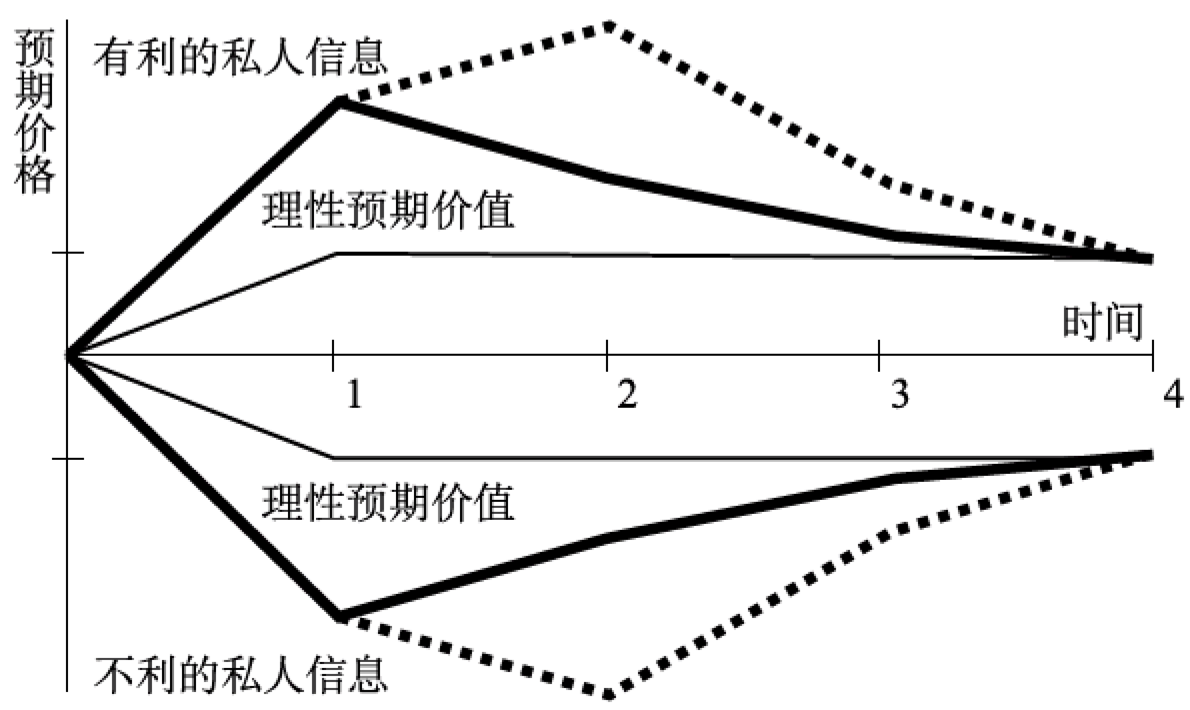


图 8.3 过度自信投资者的时间价格函数

2）信心受结果影响的情况

DHS 模型进一步讨论了投资者信心受结果影响的情况。来自心理学的证据表明，行为及其导致的产出结果将影响信心水平。事件的结果如果与个体的信念及行为相一致的话，将导致其自信心的强烈增加，而事件的结果如果与个体的信念和行为不相一致，由于受到有偏的自归因的影响，个体的自信心也不会有明显的减弱。

作为一个有识投资者，在起初并不过度自信，只是基于自己的私人信息进行证券买卖。假设其后的公开信息肯定了该投资者的交易，则投资者的信心将会更加强烈，而如果其后的公开信息并不肯定该投资者的交易，则投资者的信心也只会略有降低或保持不变。这意味着，公开信息将增加自信程度而加强过度反应，连续的过度反应将导致在最初的过度反应阶段产生正的自相关性。当公开信息重复到达，从而推动价格回归基本面，最初的过度反应将在长期反转。

将上述过程反映到模型中，就是图 8.3 中虚线部分(图 8.3 中的(0-1)部分和粗实线重合)。如前所述，由于有偏的自我归因，公开信息平均来说增加了投资者的信心，增加了过度反应现象。从而在含有噪声的公开信息到达的时期 2，股价并未得到修正，过度反应得到加强，进一步偏离基本价值，直到重复的公开信息将价格推回基本面。

在 DHS 模型的信心受结果影响的情况下，其结论是：如果投资者的信心改变由有偏的自归因主导，且如果过度反应和修正是逐步的，那么证券价格的变化将表现出无条件的

短期滞后的正自相关(动量)和长期滞后的负自相关(反转)。

3. HS 模型

Hong 和 Stein(1999)提出了考虑不同交易者作用机制的模型,简称 HS 模型。在 HS 模型中,两个由有限理性的投资者组成的群体进行相互作用,在此有限理性是指投资者只能处理全部可得信息的子集。HS 模型将两个群体分别称为“消息观察者”和“动量交易者”。

“消息观察者”不依赖于过去的价格信息,而是利用有关未来价值的私有信息进行预测。而“动量交易者”则完全依赖最近的价格变化。模型假设私有信息在“消息观察者”中的扩散是缓慢的,也即具有反应不足的倾向,从而在股价上产生动量,此时“动量交易者”察觉到动量,进而进行交易套利,将价格推向极致。

HS 模型的分析主要分为两部分:一是在只有“消息观察者”的情况下,价格对新的信息调整缓慢,在此情况下只存在反应不足而没有过度反应。其原因在于“消息观察者”并不从价格中推断信息,且信息在“消息观察者”中扩散是缓慢的。二是加入“动量交易者”的作用,其以观察过去的价格为条件,利用“消息观察者”造成的反应不足进行套利。在存在有效的风险容忍度的情况下,可以期望其行为将促使市场接近有效。然而,在“动量交易者”只是受限于使用简单的动量策略的情况下,上述直觉判断是不正确的。

当“动量交易者”只能在时期 t 根据时期 $t-2$ 和时期 $t-1$ 的价格变化进行交易时,其利用“消息观察者”引起的反应不足进行套利的结果,将导致最初的向基本面变化的价格最终加速,形成对消息的过度反应,在“动量交易者”是风险中性时尤其如此。

在 HS 模型中,“消息观察者”和“动量交易者”之间存在相互作用的交易机制。当“消息观察者”观察到利好消息时,即进行买入行动,最初的买入产生的价格动量较小,在某种程度上可理解为反应不足。其后,部分“动量交易者”观察到起初的价格动量后立即跟随进行买入,从而引起价格产生更大的变化(动量),对后来的“动量交易者”施加了负面影响,使他们错误地理解为有更多的利好消息到达,导致越来越多的“动量交易者”加入,过多的买入力量推动价格远远超过利好信息所能引起的应有涨幅,产生价格的过度反应,最终价格趋势反转下跌。当“消息观察者”观察到的是利空消息时,价格将向下过度反应,最终反转上升。在 HS 模型中“动量交易者”作为噪声交易者,在其与“消息观察者”的相互作用下,导致价格的反应不足和过度反应。

8.4 过度反应的实证检验

最早提出过度反应的是德博特和塞勒(1985),他们发现美国股票市场存在着过度反应现象。随后,很多学者对过度反应是否存在以及过度反应产生的原因进行了研究。从研究对象来看,一些研究试图检验整个市场是否存在过度反应现象,这类研究多数是参考德博特和塞勒的研究方法,构造输家赢家组合,计算累积超额收益率,再进行分析。沈艺峰和吴世农(1999)就是通过这种方法进行研究的,他们发现我国股票市场不存在过度反应现象。Chopra 等(1992)基于股票以往 5 年的收益率构造赢家组合和输家组合,发现在

随后的 5 年输家组合比赢家组合的收益率高，这证明过度反应现象存在，他们还发现小公司的过度反应现象比大公司严重。

另外一些有关过度反应的研究是针对某类特定的事件，这些研究通常使用事件研究法(event study)，检验某类特定事件是否引起了市场的过度反应。事件研究法由 Ball 和 Brown(1968)以及法玛等(1969)开创，其原理是根据研究目的选择某一特定事件，研究事件发生前后样本股票收益率的变化，进而解释特定事件对样本股票价格变化与收益率的影响，主要被用于检验事件发生前后价格变化或价格对披露信息的反应程度。

事件研究法是基于有效市场假设的，即股票价格反映所有已知的公共信息，由于投资者是理性的，投资者对新信息的反应也是理性的，因此，在样本股票实际收益中剔除假定某个事件没有发生而估计出来的正常收益(normal return)就可以得到异常收益(abnormal return)，异常收益可以衡量股价对事件发生或信息披露的异常反应程度。

股票的正常收益和异常收益可根据 CAPM 模型计算出来：$R_{i,t}=\alpha_i+\beta_i R_{m,t}+\varepsilon_{i,t}$。

$R_{i,t}$表示 i 公司 t 期的报酬率，$R_{m,t}$表示 t 期的市场加权指数股票报酬率，股票的收益可以分为两部分：$\alpha_i+\beta_i R_{m,t}$代表正常收益；$\varepsilon_{i,t}=R_{i,t}-(\alpha_i+\beta_i R_{m,t})$代表异常收益。基于有效市场假设，$E(\varepsilon_{i,t})=0$；若 $E(\varepsilon_{i,t})\neq 0$，则说明股价对事件异常反应。

用来检验市场过度反应的比较常用的事件是盈余公告、分析师评级调整等。Zarowin(1989)检验了股票市场是否对极端的盈余信息过度反应，研究发现在极端盈余年度之后的 36 个月里，利润低的公司比利润高的公司股票收益高，但这不意味着股票市场对盈余信息过度反应，因为他们发现那些利润低的公司多数是小公司，股票收益的差距是由公司规模而不是过度反应引起的。Brown 等(2014)研究发现分析师对公司评级调高会引起共同基金羊群般涌入市场，而分析师对公司评级调低会引起共同基金集体抛售股票，分析师对共同基金羊群行为的影响在评级调低时更大，这引起了股票市场的过度反应。谭洪涛等(2011)对公允价值与股市过度反应进行了实证研究，研究发现股市过度反应与公允价值变动损益显著相关，我国金融业上市公司的金融资产市值计量模式显著放大了股市泡沫，且在不同的市场条件下，股票回报对公允价值的过度反应程度呈现出非对称性。

延伸阅读 8.1　个人投资者和机构投资者的不同反应模式

市场上存在着不同类型的投资者，由于投资者在信息获取和处理上的差异，不同类型的投资者对信息的反应也有所不同。机构投资者在信息资源和处理能力上优于个人投资者，其交易行为也相对理性。因此，研究个人投资者和机构投资者的不同反应模式具有现实意义。

游家兴(2008)研究了中国股市个人投资者和机构投资者的信息反应模式，发现在短期内，机构投资者对信息的反应相对理性，而个人投资者反应过度，这也导致了整个市场的过度反应；在中长期内，个人投资者逐渐走向理性，而机构投资者对新信息存在一个反馈调整的渐进过程，在其主导下，市场整体表现出反应不足。

这样的研究结果支持了机构投资者的参与有助于信息传递和市场稳定的观点。未来

这方面还将有更多、更深入的研究。

资料来源:游家兴.谁反应过度,谁反应不足——投资者异质性与收益时间可预测性分析[J].金融研究,2008(4):161-173.

8.5 交易制度对过度反应的影响

过度反应现象在股票市场很常见,为了市场稳定,监管者设立了一些交易制度来抑制过度反应,涨跌幅限制就是其中一种。设立涨跌幅限制的目的是使投资者有足够的时间来消化市场情绪,合理分析市场信息,进而作出理性的投资决策。从 1996 年 12 月 16 日开始,我国沪深证券交易所对所有上市股票(含 A、B 股)及基金交易实行涨跌幅限制,规定在一个交易日内,每只证券的成交价的涨跌幅度不得高于或低于前一日收市价的 10%。

涨跌幅限制在我国实施以来,有关涨跌幅限制作用的争议很大。一种观点认为,涨跌幅限制起到了助涨助跌的作用,主力机构大户利用股价涨跌停来诱发过度反应或者增大过度反应的程度。而相反观点认为涨跌幅限制给投资者更多时间对信息进行分析,能够减少过度反应。吴林祥等(2003)的研究发现在股价发生涨跌停时,虽然涨跌幅限制不能完全消除过度反应,但是可以显著减小过度反应的程度,价格涨跌停本身并不是导致过度反应的原因,因为股价不仅在涨跌停时存在过度反应,而且在大幅度变化但未达到涨跌停时也存在过度反应。

除了涨跌幅限制,熔断机制也是一种为了抑制股票市场的过度反应而设立的机制,1987 年美国股灾后,美国证券交易委员会提出设立熔断机制,该机制规定以某一指数上下浮动一定比例为触发水平,一旦市场下跌或上涨至触发水平,交易就暂停一段时间。熔断机制旨在为投资者的非理性行为提供一个“冷却时期”,引导投资者对证券价值进行重估。

2015 年我国股票市场经历了巨幅震荡,为了维护金融市场的稳定,2016 年 1 月 1 日,我国开始实施熔断机制,该机制以沪深 300 指数为标的,规定当指数触发 5%熔断阈值时,市场暂停交易 15 分钟,当指数触发 7%时,全天交易暂停。在熔断机制实施的第一个交易日,即 2016 年 1 月 4 日,股市开盘即下跌,5%和 7%熔断阈值先后被触发。1 月 7 日熔断机制再度触发,开盘 15 分钟就收市。熔断机制被舆论认为是加速股指下跌的原因,在巨大的争议中,证监会宣布从 1 月 8 日起暂停实施指数熔断机制。

熔断机制发挥稳定市场作用的主要机理在于投资者能够冷静下来,利用交易暂停时间去了解真实信息,合理地对股票进行估值,减少恐慌性卖出行为。然而,在我国股市熔断触发导致交易暂停的期间,监管当局和上市公司并未发布新的信息,引导投资者重估股票价值,因而冷却效应的机制难以实现。与冷却效应相反的磁力效应则认为熔断机制会加剧股市的下跌,其主要机理在于投资者担心交易暂停后交易无法实现,倾向于在熔断之前就提交卖出订单,引发市场恐慌情绪,加速市场崩盘。杨晓兰和金雪军(2017)的研究发现,在沪深 300 指数下跌至熔断临界值之前,熔断机制会加剧卖单的大量涌现,存在显著的磁力效应,在股指下跌的过程中,熔断机制的磁力效应存在从沪深 300 成分股向非成分股的溢出,具体见延伸阅读 8.2。

延伸阅读 8.2　熔断机制的磁力效应

杨晓兰和金雪军(2017)利用自然实验的思路,以 2016 年 1 月 4 日和 7 日为熔断组,以 2015 年沪深 300 指数跌幅超过 7%的 3 个交易日为控制组,检验熔断机制对股票市场产生的影响。研究结果表明在指数下跌至熔断临界值之前,熔断机制会加剧卖单的大量涌现,导致市场订单流不平衡性提高,存在显著的磁力效应。

考察熔断机制和涨跌停板的交叉效应,发现在指数下跌的初期,对于已经跌停的股票,涨跌停板和熔断机制不会产生叠加效应,而对于尚未跌停的股票,熔断机制产生了显著的作用;当指数下跌 5%～7%,熔断机制和涨跌停板制度存在叠加效应,订单流不平衡性进一步恶化。熔断机制的磁力效应存在从沪深 300 成分股票向非成分股票扩散的趋势。此外,股指期货交易、融券交易、沪港通净流入能缓解熔断机制的磁力效应;而融资交易余额越大,熔断机制刺激卖盘涌现的程度越高。

资料来源:杨晓兰,金雪军. 我国股票市场熔断机制的磁力效应:基于自然实验的证据[J]. 金融研究,2017(9):161-177.

本章小结

(1)反应不足是指除最初的反应后,股价的平均收益在利好消息到达的一段时间内比利空消息到达的同样时间内要高的现象。过度反应是指随着利好消息的到达后,在 $t+1$ 期股价的平均收益低于利空消息到达后的平均收益的现象。

(2)投资者在现实中受到多种偏差因素的影响,如代表性偏差、可得性偏差、锚定和调整、保守主义、损失厌恶、过度自信与自归因偏差、有限理性、后悔心理、信念坚定等,但投资者究竟受到何种因素的影响以及多大程度的影响,是与投资者个体不同的特质以及其所处的不同市场环境密切相关的。

(3)BSV 模型可以用来解释股价在短期内的惯性效应和长期的反转效应。短期:受保守主义影响,表现为反应不足;长期:经历一系列的正盈利冲击,投资者会调整自己的保守特征,利用代表性法则推断盈利有增长趋势,不断推高股价,最终股价产生反转。

(4)DHS 模型着重考虑投资者在阐释私有信息而不是公开信息时存在的偏差,该偏差主要有过度自信和自归因偏差。

(5)HS 模型假设私有信息在“消息观察者”中的扩散是缓慢的,也即具有反应不足的倾向,从而在股价上产生动量,此时“动量交易者”察觉到动量,进而进行交易套利,将价格推向极致。

(6)事件研究法是根据研究目的选择某一特定事件,研究事件发生前后样本股票收益率的变化,进而解释特定事件对样本股票价格变化与收益率的影响,主要被用于检验事件发生前后价格变化或价格对披露信息的反应程度。

(7)涨跌幅限制和熔断机制都是监管者为抑制市场过度反应设立的机制,然而这些制度实施以来,对于它们是否能起到抑制过度反应的作用还存在争议。

思考题

1. 如何界定反应不足和过度反应？利用过度反应现象可以构造怎样的交易策略？
2. 除了 BSV 模型、DHS 模型和 HS 模型，你还知道哪些过度反应的理论模型？
3. 运用事件研究法进行研究设计时需要注意哪些问题？

第9章

投资者情绪

引导案例：从搜索行为看投资者情绪

近年来，越来越多的学者关注到投资者情绪与资本市场的关系。随着计算机和互联网技术的飞速发展，学者们从互联网提供的大数据中提取能反映投资者情绪的信息。Da等(2015)利用网民对“衰退”“失业”“破产”等与投资者恐慌情绪相关的词语在谷歌上的搜索量，构建了FEARS(恐慌)指数。FEARS指数反映了投资者的恐慌情绪。

由图9.1和图9.2可知，2008年金融危机爆发后“recession”(衰退)、“bankruptcy”(破产)搜索量剧增，这表明金融危机的爆发引起了投资者恐慌情绪的爆发，当然投资者情绪与资本市场的关系绝不是单向的，二者互相反馈。当资产价格走高时，乐观的投资者情绪会推动价格上涨；当资产价格下降时，悲观的投资者情绪会加剧市场的崩盘。

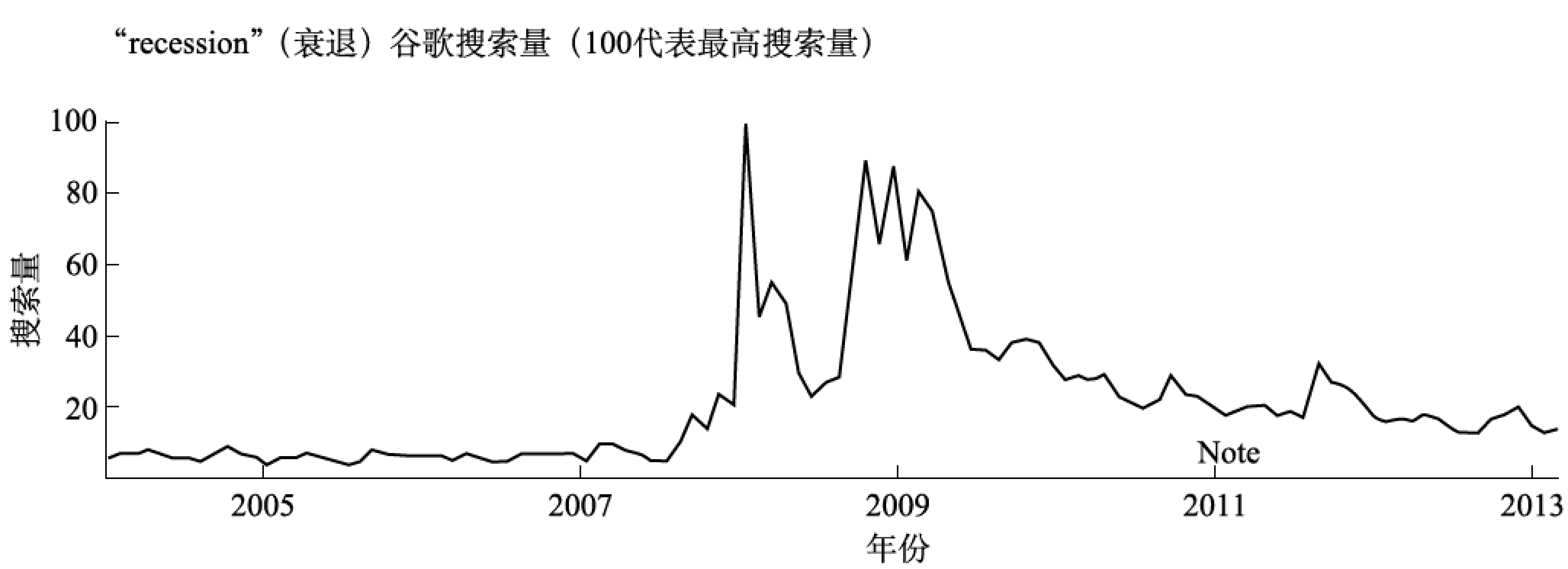


图9.1　2004—2013年“recession”(衰退)在谷歌上的搜索量

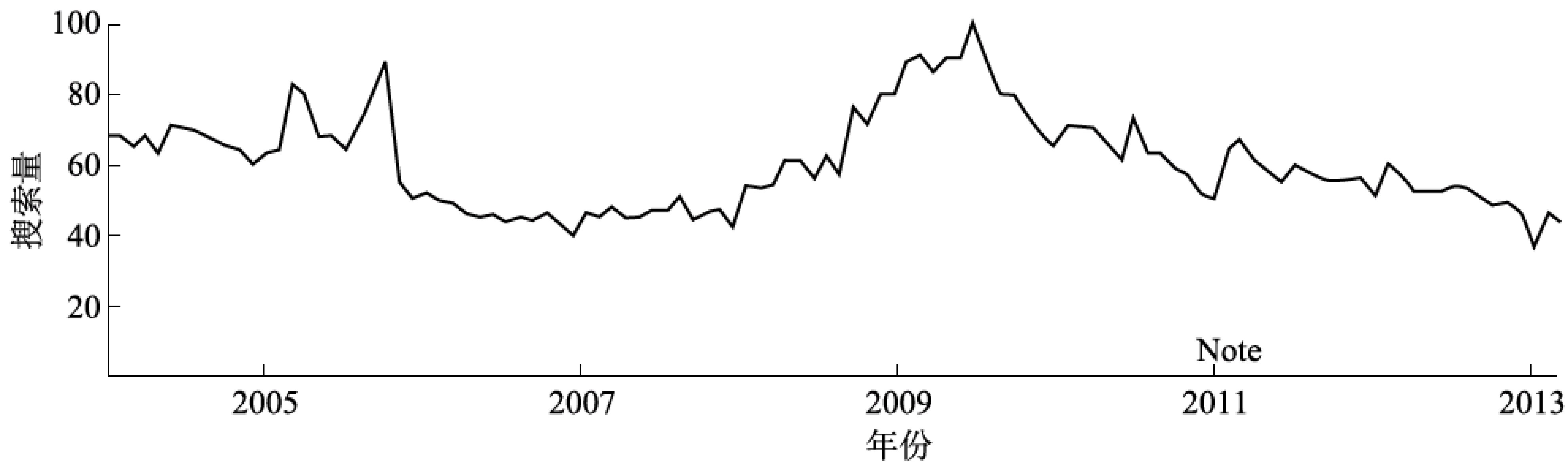


图 9.2 2004—2013 年"bankruptcy"(破产)在谷歌上的搜索量

资料来源:DA Z,ENGELBERG J,GAO P. *The sum of all FEARS investor sentiment and asset prices*[J]. *The review of financial studies*,2015,28(1):1-32.

本章将详细介绍投资者情绪的含义、度量方法及其影响股票市场的理论分析和实证检验结果。

9.1 投资者情绪的含义

投资者情绪是行为金融学中的基本概念,学术界始终没有对其定义达成共识。Lee 等(1991)将投资者情绪定义为无法被基本面因素所解释的收益率预期。Baker 和 Stein(2004)认为投资者情绪反映了投资者的价值判断与资产真实价值的偏差。Baker 和 Wurgler(2006)提出了两种投资者情绪定义:第一种,投资者情绪是指投资者的投机倾向;第二种,投资者情绪是对股票市场整体的乐观与悲观心态。

心理学的研究表明,情绪会影响人们的决策行为,而且这种影响很多时候不被自身察觉。投资者情绪主要由投资者的偏好、情感、从众心理等形成,投资者情绪会影响其选择偏好,进而影响其投资决策,投资者的投资决策会影响股票价格,而股票价格又会反馈到投资者的情绪中。

在投资者情绪的相关研究中,De Long 等(1990)提出的 DSSW 噪声交易者模型具有较大影响力。该模型表明:在有限套利的环境中,如果投资者情绪相互影响,套利者将无法消除非理性行为导致的错误定价,投资者情绪会成为影响金融资产均衡价格的系统性风险。Lee 等(1991)在对封闭式基金的考察中发现基金的持有者中有些是噪声交易者,这些交易者对未来收益的预期很容易受到自身乐观态度或悲观态度的影响,出现相对于基金资产净值的溢价或折价现象,噪声交易者情绪的波动可以解释封闭式基金折价之谜。

投资者情绪是影响股票市场表现的重要因素。当市场持续上涨时,投资者情绪高涨,表现得过于激进;当市场持续下跌时,出于对股票市场未来发展的担忧,投资者情绪十分低落,表现得过于保守,甚至过于恐惧。因此,受到投资者情绪影响,投资者的决策行为不是完全理性的,最终导致股票市场整体出现上涨或下跌现象。

金融市场上出现的一些异象无法用传统金融学理论解释,已有一些学者从投资者情绪角度解释诸如封闭式基金折价、IPO 抑价等市场异象。也有一些学者分析了投资者情

绪对不同股票影响的差异性。有关投资者情绪的研究很多，学者们采用不同的方式去度量投资者情绪，本章第 9.2 节将对此详细介绍。投资者情绪影响股票市场的理论分析和实证检验将会在第 9.3 和第 9.4 节展开介绍。

专栏 9.1　恐惧贪婪指数

投资者在进行投资决策时常常被情绪驱动，恐惧和贪婪能够影响投资决策，恐惧会引发抛售，而恐惧情绪的蔓延会导致股价崩盘，贪婪会推动股价的不断攀升，使得股价远高于内在价值，产生资产泡沫。美国有线新闻财经网(CNN Money)发布了恐惧贪婪指数(Fear & Greed Index)来反映市场情绪，该指数基于七个指标：股价动量、股价强度、股价宽度、看涨看跌期权、垃圾债券需求、市场波动性、股票收益与国债收益的差额，这些指标加总平均得到一个 0～100 之间的值，值越高，说明市场情绪越贪婪，值越低，说明市场情绪越恐惧。

恐惧贪婪指数在 2008 年金融危机时期降到 12，这反映了市场陷入极度恐惧之中，随后市场回暖，恐惧贪婪指数上升。近年来美国股市处于牛市，恐惧贪婪指数经常飙升到 50 以上，说明投资者处于贪婪状态。恐惧贪婪指数与市场行情紧密相关，是一个比较好的反映投资者情绪的指标。

资料来源：http://money.cnn.com/data/fear-and-greed/.

9.2　投资者情绪的度量方法

越来越多的研究者关注到投资者情绪对资本市场的影响，研究结果却不尽相同，这很大程度上是因为学者们用不同的方式度量投资者情绪。度量投资者情绪的方法主要有三种：一是通过对投资者的直接调查构建情绪指标，如通过问卷调查考察投资者信心等；二是通过市场表现间接度量投资者情绪，如将交易量、封闭式基金折价、IPO 发行量及首日收益等指标作为投资者情绪的代理变量，目前的研究多是采用基于若干单一指标的综合性指标来度量投资者情绪；三是基于搜索行为、文本挖掘等方法构建的投资者情绪的新指标。下面将详细介绍这些投资者情绪的度量方法。

1. 直接指标

度量投资者情绪的直接指标指的是通过问卷调查等方式得到的投资者对市场未来的预期和看法。比较常用的直接指标有美国个人投资者协会(American Association of Individual Investors，AAII)指数，该指数反映了参与问卷调查的会员对股票市场持有牛市、熊市和中性观点的比例，例如，耶鲁大学编制的投资者信心指数，美国密歇根大学调查研究中心(Survey Research Center，SRC)编制的消费者信心指数以及 UBS(瑞士联合银行集团)和 Gallup(盖洛普咨询公司)编制的投资者乐观指数。中国反映投资者情绪的直接指标有国家统计局发布的中国消费者信心指数、基于投资者对未来走势判断的央视看盘指数、《股市动态分析周刊》发布的好淡指数。好淡指数将投资者对股市的情绪分为“好”和“淡”，用看涨投资者与总投资者之比来构造指数。

这些直接指标能够直接反映投资者的心理特征，但是存在一些问题：首先，投资者在问卷调查中可能会有顾虑，直接调查的结果与真实的投资者心理可能会有偏差。其次，在实际的投资决策中，投资者情绪对决策的影响因人因时而异，Fisher 和 Statman（2000）研究发现，虽然投资者对未来看涨或看跌，但是在实际投资行为中，他们并不会完全按照情绪行事，或者说投资者在其实际投资行为过程中要比在情绪的变化上更明智。此外，直接调查的结果还会受到调查样本的限制，导致指数有偏。

因此，通过直接度量获得的投资者情绪指数虽然可以直观地反映投资者的情绪，但它并不能全面反映投资者在决策过程中真实的情绪，这就需要从其他方面来考察投资者情绪，如通过一些基于市场表现的指标间接衡量投资者情绪。

2. 间接指标

度量投资者情绪的间接指标很多，目前学术界使用较多的有交易量、封闭式基金折价、IPO 发行量及首日收益、共同基金净赎回、波动率指数（volatility index，VIX）等指标。还有一些学者用换手率、腾落指数[①]、投资者开户增长率、中签率等指标考察投资者情绪。

目前使用较多的方法是运用主成分分析法构建基于若干单一指标的综合指标，这样能够较为全面地反映投资者情绪。构建综合指标的代表是 Baker 和 Wurgler（2006），他们基于封闭式基金折价、交易量、IPO 数量、上市首日收益、股利收益和股票发行占总发行比例六个单项情绪指标构建了一个综合指标（BW 指数），同时控制了相应的宏观经济变量（如工业生产指数、经济景气指数以及耐用/非耐用/服务消费增长等）。BW 指数被广泛应用于有关投资者情绪的研究中。易志高和茅宁（2009）在 BW 方法的基础上，融入能反映国内股票市场投资者情绪变化的指标，即封闭式基金折价、交易量、IPO 数量及上市首日收益、消费者信心指数和新增投资者开户数等，同时控制了宏观经济变量，构建了中国股票市场投资者情绪综合指数（CICSI）的月度指标。此外，国内的其他学者也基于 BW 方法选取了不尽相同的指标，构建投资者情绪综合指标，如魏星集等（2014）基于 BW 方法选取封闭式基金折价率、IPO 数量及上市首日收益率、新增开户数、市场换手率和消费者信心指数，构建 ISI 投资者情绪指数。目前投资者情绪指数越来越多地被研究机构使用，国泰安 CSMAR（中国股票市场研究）数据库已推出中国投资者情绪指标研究数据库，其中收录了 BW 综合情绪指数主要因子、CICSI 和 ISI 月度指数，以及其他能反映投资者情绪的指标。如图 9.3 和图 9.4 所示。

由中国投资者情绪综合指标 ISI 可以看出，在 2007 年、2008 年中国投资者情绪急剧上升后又急剧下降，在 2015 年也发生了投资者情绪急剧变化的类似情况，结合我国 A 股市场，2008 年和 2015 年正是两次股灾的时间，投资者情绪的剧烈变化与股票收益的剧烈变化相互关联，另外，这也说明 ISI 代表投资者情绪具有一定的可信性。CICSI 虽然不像 ISI 那样波动剧烈，但是走势与 ISI 大致相同，这说明 CICSI 也能在一定程度上代表投资者情绪。

---

① 腾落指数是指股票每天上涨与下跌家数之差，月度值为交易日累计值。

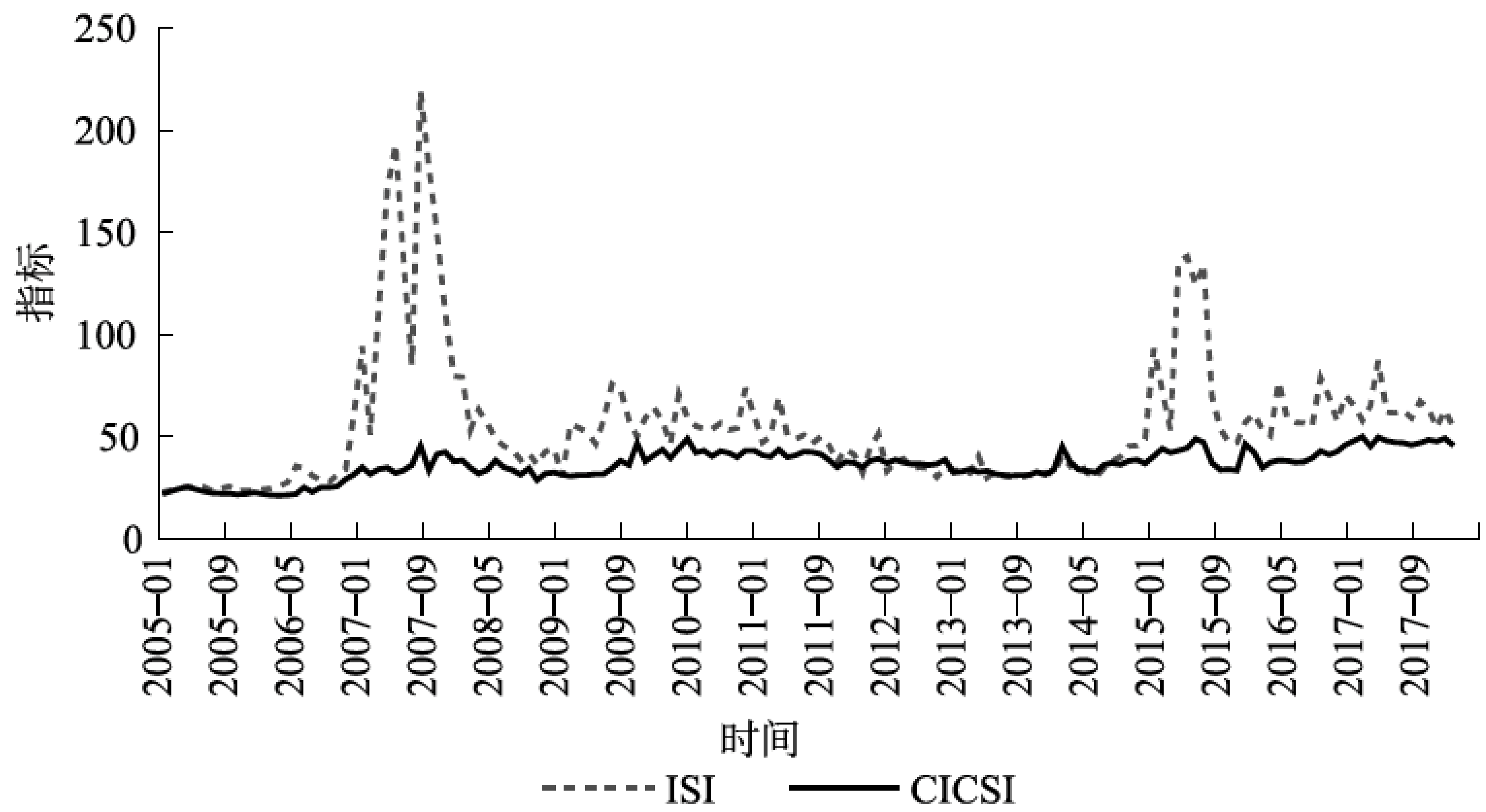


图 9.3　2005 年 1 月—2017 年 12 月中国投资者情绪综合指标 ISI 和 CICSI

资料来源：此图根据国泰安 CSMAR 数据库提供的 2005 年 1 月—2017 年 12 月中国投资者情绪综合指标 ISI 和 CICSI 月度数据绘制而成。其中 ISI 指数参考文献：魏星集，夏维力，孙彤彤．基于 BW 模型的 A 股市场投资者情绪测度研究[J]. 管理观察，2014(33)：71—73。CICSI 指数参考文献：易志高，茅宁．中国股市投资者情绪测量研究：CICSI 的构建[J]. 金融研究，2009(11)：174—184。

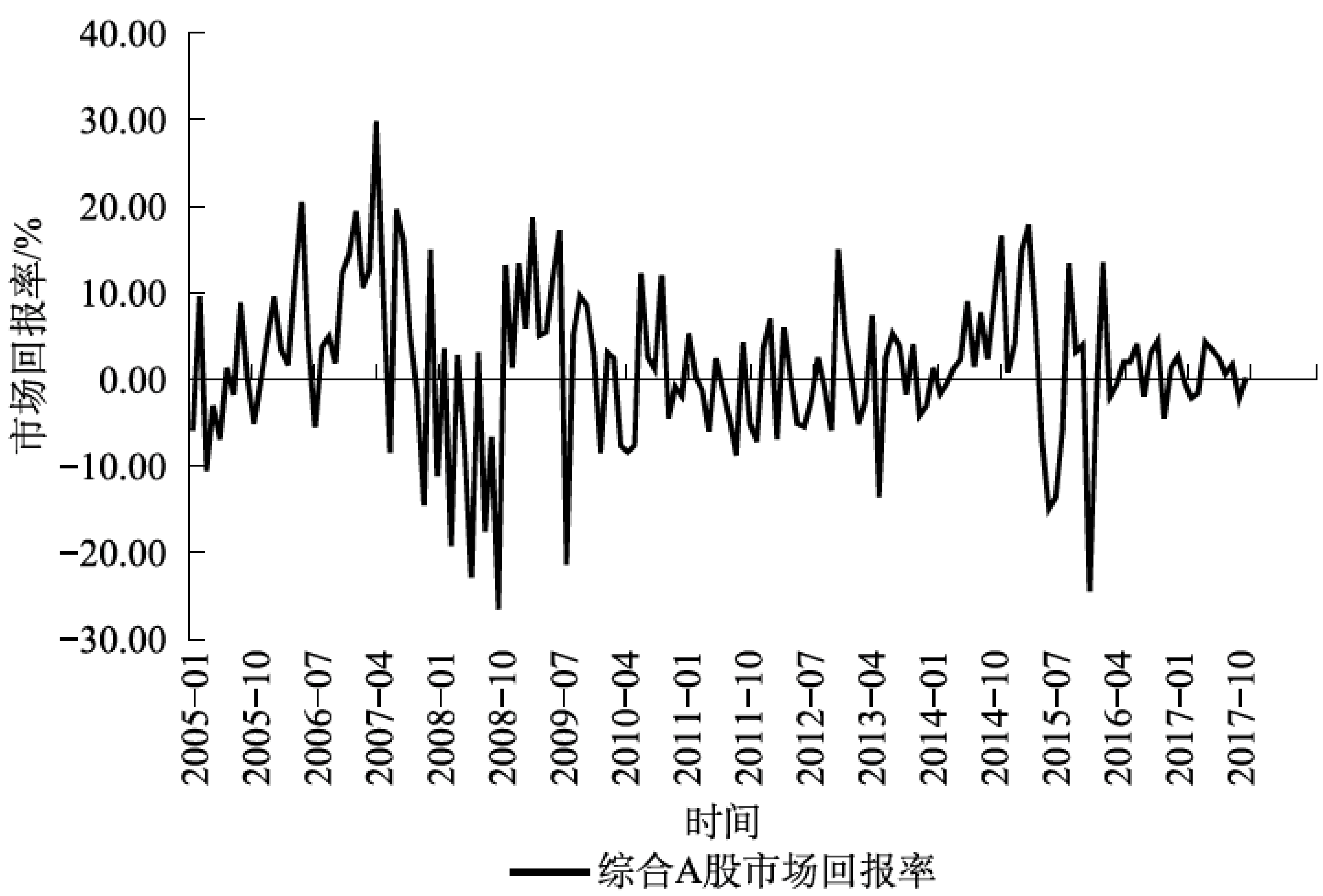


图 9.4　2005 年 1 月—2017 年 12 月中国综合 A 股市场月回报率

资料来源：此图根据国泰安 CSMAR 数据库提供的 2005 年 1 月—2017 年 12 月综合 A 股市场流通市值加权月回报率(考虑现金红利再投资)绘制而成。

这些间接指标比较客观，并且相比于直接指标可得性更高，但是市场表现受多种因素的影响，无法把投资者情绪从诸多因素中分离出来，因此寻找更合理的反映投资者情绪的指标仍是学术研究的热点。

专栏 9.2　国信市场情绪指标 GSSI

国信证券经济研究所选择具有日数据的如下七个指标构建市场情绪综合指标 GSSI：分析师一致预期分歧指数、平均换手率、主卖成交量/主买成交量、相对强弱指标、资金流量指标、银行间同业拆借利率(7 天加权)、标普 500 日收益。

其中分析师一致预期分歧指数是由个股分歧指数经过流通市值加权计算得到，而个股

分歧指数是指 3 个月内研究员预测的 EPS(每股盈余)数据的标准差/均值。平均换手率指的是过去 10 天平均换手率/过去 250 天平均换手率。主卖成交量/主买成交量指的是每日主动卖出成交量/每日主动买入成交量,反映了市场看涨看跌力量的对比。相对强弱指标指的是一段时期内上涨点数和涨跌点数之和的比,反映了市场上多空双方买卖力量的强弱程度。资金流量指标是成交量的相对强弱指标,根据成交量来反映市场供需关系和买卖力道。

基于以上指标对样本期的股票进行主成分分析,最后得到衡量市场情绪的综合指标 GSSI,并利用 GSSI 对市场走势进行预测。

资料来源:《数量化投资技术系列报告之四十六:国信市场情绪指针 GSSI 构建与应用》,国信证券经济研究所。

3. 基于互联网与大数据的新指标

随着计算机技术的飞速发展和互联网的广泛使用,个体投资者的相关数据可得性提高,越来越多的研究者开始基于互联网与大数据构建衡量投资者情绪的新指标。目前使用较多的是基于媒体报道、社交论坛等文本信息挖掘的情绪指标,还有一些基于搜索行为的情绪指标。

目前一般使用文本分析(textual analysis)方法从文本大数据中提取情绪信息。文本分析是指通过对文本内容进行挖掘和数据分析,获得文本提供者的特定立场、观点、价值和利益,并由此推断其意图和目的。[①] 从文本大数据中提取的情绪一般是指文本中体现的积极或消极的情感倾向。目前经常使用的文本分析方法是“文字包”(bag of words)技术。它在假设文字顺序并不影响信息传递质量的前提下,将大量的文本内容简化成一个以不同文字为行、文字频数为列的矩阵。[①]“文字包”技术包括词汇分类字典法、文本词汇加权法和基于机器学习的多种分类法(如朴素贝叶斯法)。此外,研究者还可以通过不同的文本特征(诸如文本可读性、文本叙述方法等)进行文本分析,从而推断文本创作者所持的观点或情绪,并考察相应的市场反应。

Garcia(2013)用洛克伦与麦克唐纳字典(Loughran and MacDonald Work Lists)中的积极和消极词汇分类来提取 1905—2005 年《纽约时报》中金融市场和华尔街话题两个专栏中体现的情绪信息,发现构建的情绪指标能够在经济衰退时有效预测股票市场收益。Solomon 等(2014)发现投资者更愿意购买那些近期被新闻积极和正面(用 LM 词汇字典衡量)报道的基金,而忽略收益同样不错但新闻关注度不高的基金。

基于机器学习的朴素贝叶斯法在文本分析中很常用。给定一个已知的训练数据集,朴素贝叶斯法先使计算机学习该数据集中单词与其所属类别的对应关系,然后依照这个关系对其他的文本信息进行最为相似的分类。它的实质是首先利用贝叶斯条件概率公式,计算出已知文档属于不同文档类别的条件概率,然后根据最大后验假设将该文档归结为具有最大后验概率的那一类。Antweiler 和 Frank(2004)最早在金融领域使用该方法进行文本分析,他们通过雅虎金融等网络媒体分析了道琼斯工业指数中所包含的 45 家公司的近 150 万份股票信息,发现股票的新闻数量与随后的交易量存在较强联系。Huang 等(2014)从标准普尔 500 指数所包含公司的分析师报告中提取了情绪信息,发现分析师报

① 唐国豪,姜富伟,张定胜. 金融市场文本情绪研究进展[J]. 经济学动态,2016(11):137-147.

告中的正向情绪对公司未来5年的盈利增长有很好的预测价值。

国内学者游家兴和吴静(2012)从报道基调、曝光程度、关注水平三个维度,用文本分析法构建了衡量媒体情绪指数的综合评价指标体系。他们发现当新闻报道所传递出的媒体情绪越高涨或越低落时,股票价格越有可能偏离基本价值水平。徐巍和陈冬华(2016)借助文本分析技术,研究了上市公司官方微博的信息作用特点,发现上市公司微博信息披露可以带来当日的超额回报与交易量的显著增加。杨晓兰等(2016)运用文本挖掘技术分析东方财富网股吧针对创业板公司的帖子,构建投资者情绪指标,具体见延伸阅读9.1。

延伸阅读 9.1　东方财富网股吧反映的投资者情绪

杨晓兰等(2016)运用文本挖掘技术从东方财富网股吧中提取信息,构建投资者情绪指标。他们利用网络爬虫程序从东方财富网股吧抓取了网页文本信息。经过筛选,最终获得90多万条主帖的文本内容。借助文本挖掘工具Weka中的KNN算法将全部90多万条帖子照其所体现的情绪分成三类:“积极”“中立/噪声”“消极”。

他们参考Antweiler和Frank(2004)的处理方法,将t日某只股票i论坛中所有的情绪为“积极”的帖子的数量记为$M_{buy,i,t}$,所有的“消极”帖子的数量记为$M_{sell,i,t}$。t日股票i的情绪指数变量Sentiment计算公式如下:

$$\text{Sentiment}_{i,t} = \ln[(1+M_{buy,i,t})/(1+M_{sell,i,t})]$$

由上式可知,当积极情绪帖子数量高于消极情绪帖子数量时,该值大于零,反之则小于零。该值越大,股票市场的积极情绪越大。

上式Sentiment指数反映市场的看涨情绪,又可以称作看涨指数,由图9.5可以看出,看涨指数与创业板指数日收益率具有一定的趋同性。

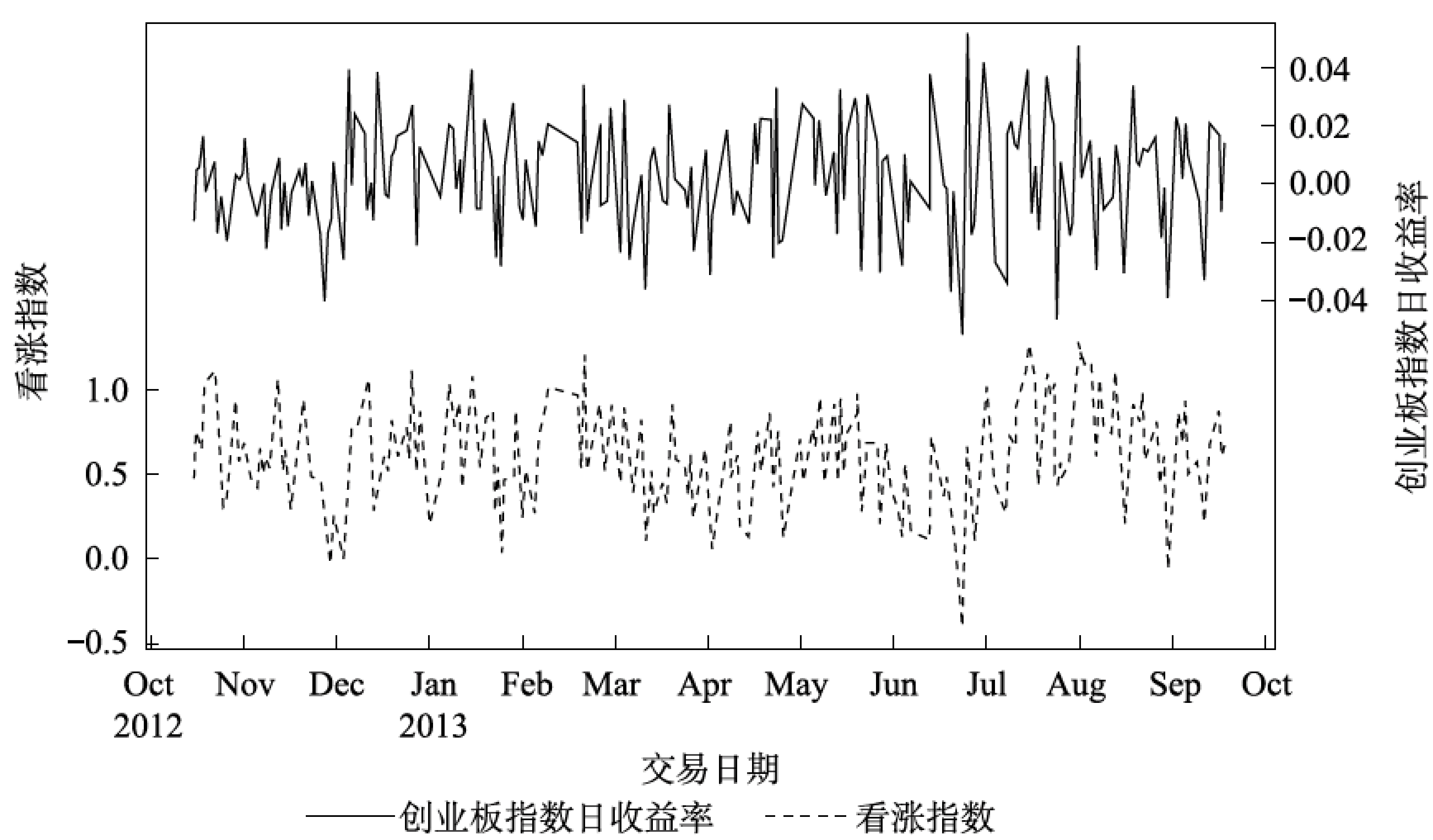


图 9.5　看涨指数与创业板指数日收益率

资料来源:杨晓兰,沈翰彬,祝宇.本地偏好、投资者情绪与股票收益率:来自网络论坛的经验证据[J].金融研究,2016(12):143-158.

除了使用文本挖掘的方法构建情绪指标，也有一些学者利用互联网搜索引擎提供的相关关键词的搜索量建立情绪指标。国外学者一般使用谷歌趋势(Google Trends)提供的关键词搜索量。Da 等(2015)利用网民对"衰退""失业""破产"等与投资者恐慌情绪相关的词语在谷歌上的搜索量，构建了 FEARS 指数。国内学者一般使用百度指数提供的关键词搜索量，目前国内学者主要用百度指数考察投资者关注，而很少有人运用百度指数研究投资者情绪。

9.3 投资者情绪影响股票市场的理论分析

投资者并不总是理性的，他们容易受到情绪影响，根据 DSSW 噪声交易者模型，市场上存在着理性投资者和噪声交易者，噪声交易者容易受到情绪影响，理性交易者难以预测未来的情绪冲击，他们的套利行为会受到限制，因此市场上的定价偏误不能被完全消除，非理性情绪会影响股票价格。另外卖空限制也是导致股票市场易受投资者情绪影响的重要因素，由于存在卖空限制，当股票价格偏离基础价值时，套利者也无法及时进行套利，这样投资者情绪会推动股价进一步偏离基础价值，导致股价的暴涨暴跌。

投资者之间的信息传递和情绪传染也会影响投资者的交易行为，如今互联网和社交平台为投资者之间的沟通交流提供了极大的便利，投资者之间传递的不仅是信息，更是情绪，投资者情绪传染容易引起市场羊群行为，导致股价的异常波动。

1. 噪声交易模型

投资者情绪研究领域最早也是最经典的理论模型是 De Long 等(1990)提出的 DSSW 噪声交易者模型。他们将股票市场上的投资者分为理性投资者和噪声交易者两类，并且假设：第一，与完全理性交易者相比，噪声交易者对市场信息的收集与处理存在差异，同时噪声交易者内部也存在显著的行为差异；但两类交易者的效用函数都包含常数形式的绝对风险厌恶系数：$U(w)=-e^{-\gamma w}$，其中，w 代表财富，γ 代表风险厌恶程度。第二，市场上存在无风险资产(每期支付固定红利 r)和风险资产(数量标准化为 1)；对于一个 t 期模型，投资者能够以 P 的成本购入 1 个单位股票，并在下一期获得回报。第三，噪声交易者在市场中占比为 n，理性交易者占比为 $1-n$；噪声交易者在 $t-1$ 期对 t 期的价格存在预期误差 ρ_t，该预期误差服从均值为 ρ^*、方差为 $\mathrm{var}(\rho)$ 的独立正态分布。此外，两类投资者在 t 期会根据其对风险资产在 $t+1$ 期的预期价格来选择资产组合，并最大化预期效用。结合上述假设，用递归方法求得风险资产的均衡价格为

$$P_t=1+n\rho^*/r+n(\rho_t-\rho^*)/(1+r)-\gamma n^2 var(\rho)/r(1+r)^2$$

上式中，第一项 1 代表股票的基本价值。第二项代表噪声交易者长期的平均估计偏差。当 $\rho^*>0$ 且其他变量不变时，表示噪声交易者对风险资产的未来价格乐观，这种对价格的压力效应将推动风险资产价格高于基本价值，噪声交易者对于风险资产的支付意愿相对较高，股价也会长期处于高位；反之，当 $\rho^*<0$ 且其他变量不变时，风险资产价格低于其基本价值，股价会长期处于低位。第三项代表由于噪声交易者 t 期的错误估价而引起的股价波动。如果 $t-1$ 期的噪声交易者对 t 期风险资产乐观的居多，则他们将推动 t 期股价

上涨。反之，则 t 期股价下降。如果 $\rho_t=\rho^*$，则意味着短期股价不会有大的波动。第四项为噪声交易者的风险溢价大小，它表示未来股市的情绪冲击对理性交易者而言是有风险的，因此理性交易者会避开高情绪的市场时期。

DSSW 模型揭示了非理性情绪会引起市场定价偏误的机理。由于市场上存在着易受情绪影响的噪声交易者，当情绪过度乐观或悲观时，理性交易者难以预测未来的情绪冲击，他们的套利行为也会受到限制。因此，理性交易者的存在也不能完全消除噪声交易者引起的定价偏误，非理性情绪会影响股票价格。

2. 卖空限制

卖空机制指的是预先借入别人的股票卖出，再在低位买回还给借方平仓来获利。我国的融券制度就是卖空机制，但我国融券业务开展时间短，能够开展融券业务的券商数量有限，而且投资者对融券业务的熟悉程度低，因此我国股票市场存在着很大程度的卖空限制，这也是我国股票市场易受投资者情绪影响的一个重要原因。

在卖空限制背景下，当股票价格偏离股票基础价值，即对未来现金流进行合理贴现后的股票基础价值，套利者无法及时准确地卖出高估值股票或买入低估值股票，因此股票价格在一定时间内、一定程度上会偏离股票的基础价值。也就是说，当股票市场中存在卖空限制时，市场的套利机制在一定程度上就失效了。

3. 投资者情绪传染

股票市场中的投资者不是孤立存在的，投资者之间的信息传递和情绪传染对于投资者的交易行为有很大影响，进而影响到股票市场的表现。如今互联网和社交网络普及，投资者经常通过各种网络平台交流市场信息和自己的观点，此时投资者情绪就会具有传染特征，像病毒一样在投资者之间扩散和传播。通过网络媒介，信息传播速度很快，而这些有关市场的信息并不都是真实的，虚假的信息也会被投资者关注，投资者情绪很容易受到他人影响，在这种环境下，投资者情绪的传染更容易引起羊群行为，导致股价暴涨暴跌、市场异常波动。

9.4 投资者情绪影响股票市场的实证检验

投资者情绪对于股票市场的影响效应主要表现在两个方面：一是投资者情绪对于股票市场所产生的系统性影响即总体效应，二是投资者情绪对不同股票的差异性影响即横截面效应。

1. 总体效应

目前学术界有关投资者情绪对股票市场的影响的研究多数集中于投资者情绪对股票市场收益率的预测作用。学者们在研究情绪对股票收益的影响时，所遵循的内在逻辑大多是投资者过分乐观或悲观的非理性信念驱动股票价格偏离基础价值，而在中长期这种过度乐观或过度悲观的情绪会被纠正，使得错误定价的股票随后向基础价值修正。即短期高涨的情绪会推动股票价格持续走高，从而对短期的市场收益率有正向影响，但是长期

这种非理性情绪将被纠正，情绪对长期的市场收益率表现出反向影响。

一些学者的实证研究验证了投资者情绪与短期收益率正相关，与长期收益率负相关。Brown 和 Cliff(2005)的研究发现，投资者情绪和近期市场收益是相关的，但情绪对市场短期未来收益不存在预测能力；就情绪对市场收益的长期影响而言，情绪与未来 1～3 年的收益负相关，情绪对于股票市场的长期收益具有预测力。伍燕然和韩立岩(2007)试图用投资者情绪来解释封闭式基金之谜和国内市场 IPOs 之谜，新股首日收益率和基金折价变化均反映了投资者情绪，投资者情绪对短期收益具有显著正向影响，对长期收益具有显著负向影响。蒋玉梅和王明照(2010)研究发现情绪在短期内与市场收益正相关，在长期上，股票价格会出现反转，进一步证实了投资者情绪对于股票收益的预测作用。Ben-Rephael 等(2012)以共同基金净赎回来衡量投资者情绪，发现投资者情绪与整体股票市场同期的超额收益正相关，与后期的超额收益负相关。

一些学者考察了不同投资主体情绪对股市收益的影响。Fisher 和 Statman(2000)研究了华尔街的投资决策者、投资通讯作者、个人投资者这三类投资者的情绪与股票收益的关系，发现不同类型投资者的情绪存在差异，但是都与 S&P500 股票收益显著负相关。Schmeling(2007)分别研究了个人投资者情绪和机构投资者情绪对股票收益的影响，发现机构投资者情绪是"智钱"(smart money)的代表，而个人投资者情绪是噪声交易者风险的代表，机构投资者在中期可以正确地预测市场收益，而个人投资者总是作出错误的预测。刘维奇和刘新新(2014)的研究发现机构投资者情绪可以帮助预测个人投资者情绪，反之不成立；另外，机构投资者在市场上表现得更为理性，他们的情绪能够预测后市，而个人投资者情绪不具有预测性。

专栏 9.3 投资者情绪滞后于市场表现

虽然有不少学者认为投资者情绪能够预测股市收益，但也有反对的声音。美国财经媒体 CNBC 的一篇新闻报道认为投资者情绪滞后于市场表现，并没有什么预测作用。据报道，在标普 500 下跌 2%之后，美国个人投资者协会调查显示的投资者情绪大跌。研究表明，看涨情绪与看跌情绪之差与标普 500 的周收益率相关，但投资者情绪滞后于收益率，而投资者情绪与未来的标普 500 的周收益率没有相关性。

换句话说，投资者情绪在市场下跌之后才下跌，当投资者情绪很低时，市场已经在底部将要上涨了，因此，投资者情绪可能发送了错误的信息，把投资者情绪当作预测市场的指标并不合适。

延伸阅读 9.2 基于谷歌搜索行为的全球投资者情绪

Gao 等(2016)根据谷歌搜索行为构建了 40 个国家的投资者情绪指数，他们同时度量了积极情绪和消极情绪，并且使用了每个国家当地的语言来构建情绪指标。他们对该指标进行了三种验证：第一，一个国家的情绪指标与该国在国际足球比赛取得的成绩相关；第二，同时在美国和其他地方上市的同一家公司在不同市场表现不同，这与所在市场的情绪相关；第三，对于小公司，在盈余公告附近的未来异常收益率与情绪指标负相关。这三

种验证都说明他们的情绪指标能够反映投资者情绪。

他们的研究发现投资者情绪与未来1周的股市收益率负相关，他们构建的投资者情绪指标能够作为国家层面市场收益的反向预测指标。他们的研究结论与以往学者的研究结论不同，这是因为投资者情绪指标的构建方法不同，他们的指标是周度的，比以往的情绪指标频率更高，另外他们依据谷歌搜索行为评价投资者情绪，这也是更直接的方式。有关投资者情绪的研究还有很大的扩展空间。

资料来源：GAO Z，REN H，ZHANG B. *Googling investor sentiment around the world*[J]. *Journal of financial and quantitative analysis*，2018(11). Available at SSRN：https：//ssrn. com/abstract=2725072.

2. 横截面效应

投资者情绪对不同类型的股票有不同的影响，Baker 和 Wurgler(2006)研究了美国股票市场投资者情绪对股票收益的横截面效应，他们认为投资者总是倾向于选择与其投机倾向相兼容的股票，在情绪高涨期，投机倾向较高，则更愿意持有高投机性的冒险型股票，对于股票会给予偏高的估值，高投机性股票对其基础价值的偏离程度更大；低情绪期与之相反。他们的研究发现那些市值小、成立时间短、波动性大、不盈利、不发放股利、高成长性的公司在市场情绪低落时能获得更高收益，而在市场情绪高涨时收益较低。

一些学者的实证研究结果与 Baker 和 Wurgler(2006)的研究结论相似，他们认为小公司、高成长性的公司的股票收益率更容易受到情绪影响。Lee 等(1991)研究了情绪与封闭式基金折价的关系，认为小市值股票和封闭式基金均主要为个人投资者所持有，受个人情绪的影响，小市值股票的收益率变动和基金折价变化呈正相关。Neal 和 Wheatley(1998)发现，以封闭式基金折价和共同基金净赎回作为情绪代理变量可以解释小市值股票的收益，以及预测规模溢价。Kumar 和 Lee(2006)研究发现小市值、低价格、机构持股比例低以及价值型(高账面市值比 / 低市净率)的公司股票大多是散户投资者集中持股与密集交易的股票，并且这些股票难以套利，因此其收益对情绪变化的敏感程度相对较高。

本章小结

(1) 投资者情绪有多种含义，学界给出的定义有：投资者的价值判断与资产真实价值的偏差；投资者对股票市场整体的乐观与悲观心态等。

(2) 度量投资者情绪的方法主要有三种：一是通过对投资者的直接调查构建情绪指标；二是通过市场表现间接度量投资者情绪；三是基于搜索行为、文本挖掘等方法构建的投资者情绪的新型指标。

(3) DSSW 模型将股票市场上的投资者分为理性投资者和噪声交易者两类，噪声交易者易受情绪影响，当情绪过度乐观或悲观时，理性交易者难以预测未来的情绪冲击，他们的套利行为也会受到限制。DSSW 模型揭示了非理性情绪会引起市场定价偏误的机理。

(4) 卖空限制使得套利行为受到限制，股票市场更容易受到投资者情绪影响。

(5) 投资者情绪容易受到他人影响，投资者情绪传染也是影响股票市场表现的重要因素。

思考题

1. 结合自身投资经验，谈谈情绪是如何影响投资行为的。
2. 评价不同类型的投资者情绪指标的优缺点。
3. 利用投资者情绪预测股票市场表现的理论机理是什么？你对此有什么看法？

第 10 章

金融市场羊群行为

引导案例:2015 年的股市风云

2015 年,中国股市在经历了长久的沉寂之后突然爆发,又在短暂辉煌之后一泻而下。下跌过程中,千股跌停、千股涨停的情景频频出现,震荡的剧烈程度甚至超过了 2008 年的那场股灾,让不少股民至今回想起来仍觉得心悸不已。2015 年沪指走势如图 10.1 所示。

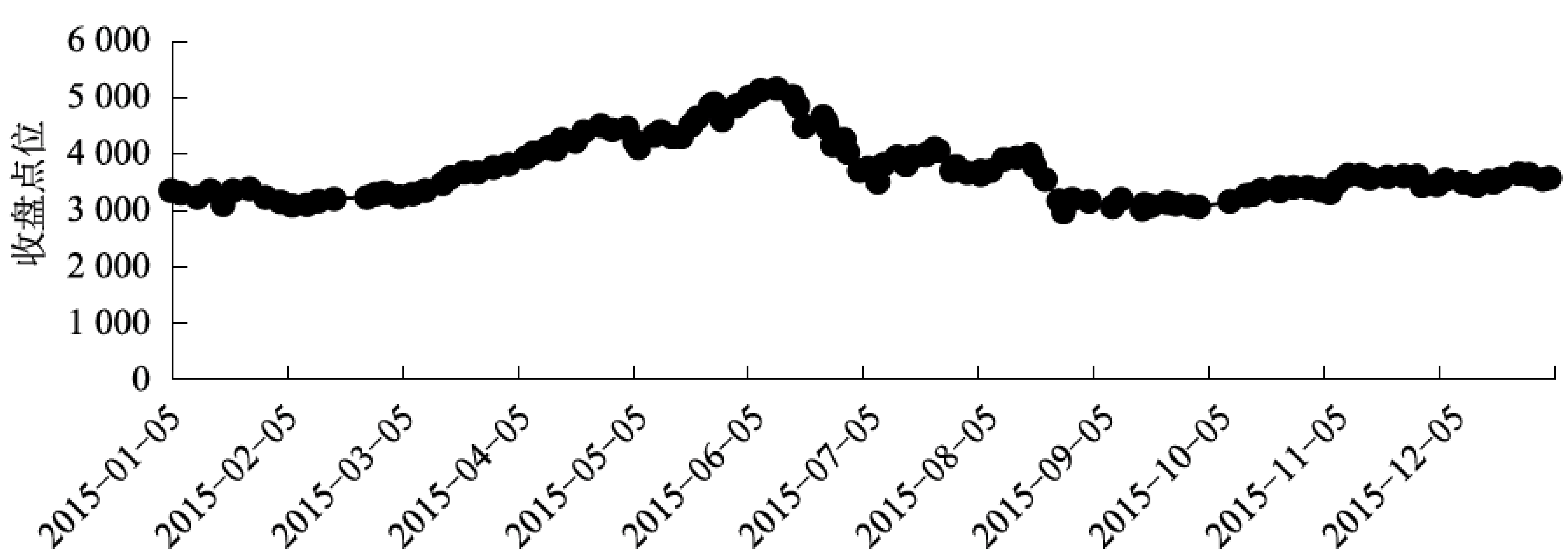


图 10.1　2015 年沪指走势

牛市初现带动全民炒股热

自 2014 年 11 月开始,沪深股市大幅上涨,让股民嗅到了久违的牛市味道。新年伊始,大盘股休整,小盘股大幅上涨。创业板指数一季度大涨 58.67%,一大批牛股涌现。如全通教育 2015 年 3 月创出 300 元的天价,创下沪深股市历史最高价纪录。

小盘股的示范效应吸引了大批中小投资者蜂拥入市,新开户数屡创新高。在政策面、资金面的共同拉动下,各种小盘股受到追捧而大幅上涨,尤其是新兴产业的股票开始一起上涨,涨幅翻倍的股票大量出现,赚钱效应扩散,一场资本市场的狂欢拉开序幕。

市场进入疯狂状态，理智无奈屈从现实

2015 年 4 月以后，市场逐渐进入疯狂状态。大批新股民开户入市，融资融券金额连创新高，新资金加上杠杆资金，使得沪深股市出现一个历史上罕见的快速上涨的杠杆牛市。到 6 月 5 日，创业板指数最高达 4 037.96 点，较 2014 年的收盘指数 1 471.76 点上涨了 174.36%。个股的疯狂程度更是史无前例。例如暴风科技，3 月 24 日上市之后持续涨停，在 31 个涨停板之后，打开涨停并继续上扬，最高达 327.01 元，较 7.14 元的发行价上涨了 45 倍。

这些疯狂的数字让仍存理智的人们开始怀疑自己的判断，进而选择向现实屈服。《中国证券报》的某篇报道引用了一位资深基金经理的话，“在资本市场，钱是最聪明的，我们做的只是尊重市场，因此，就是‘无脑买入’，也要硬着头皮买进！”

这位基金经理，和市场中许多股民一样，曾在理智与盲从间徘徊，最终还是选择了放弃思考、跟从市场。而这个市场上更多的人，则不愿思考甚至从未思考，仅仅是在媒体舆论的煽动之下、亲朋好友的劝说之下，便奋不顾身地投入股市这个疯狂的游乐场。

断崖式下跌引发股市踩踏

过快的暴涨总是隐藏着巨大的风险，2007 年的股市已经向人们证明了这一点。但沉浸在狂欢中的人们，又有谁还记得曾经的伤痛？

2015 年 6 月 18 日，上证指数下跌 3.67%，19 日下跌 6.42%。6 月 26 日，上证指数再度暴跌 7.4%，仅仅两周时间，上证指数跌去了 1 000 点。股市从天上掉到了地下，眼看之前的获利已被蚕食，高位追入的更是深度套牢，市场的情绪由贪婪转为恐惧。

股市重挫之后，尽管监管层一连串的举动态度鲜明，但似乎股市并不给面子。融资账户不断被强制平仓，加之恐慌情绪弥漫，所有人都抢着往外跑，再多的利好也只能带来短暂上扬。随之而来的，是更加疯狂的出逃。

在监管层的强势干预下，股市终于止住了迅猛跌势，上证指数维持在 3 000 点附近震荡。

回顾这短短 7 个月，人们先是蜂拥入场，后又落荒而逃，在这大起大落背后，究竟是什么力量在主导着人们产生近乎疯狂而又如此一致的行为？在本章中，就将探讨金融市场中的一种群体性行为——羊群行为。探究羊群行为的内在动因，有助于解释金融市场中各类追涨杀跌、剧烈波动的现象。

10.1 羊群行为的含义

羊群行为(herd behavior)，原意是指牛、羊等动物成群行动的现象，后来被引申来描述人类社会现象，指与大多数人一样思考、感觉、行动，与大多数人保持一致。其也称从众行为、群体行为。

羊群行为体现在社会生活中的方方面面。在餐厅点菜时，人们常会选择人气最高的菜；在选购衣服时，人们会参考当下的流行趋势；甚至是在填报高考志愿、选择未来就业方向时，人们亦会考虑当前最火爆的专业是什么、最热门的行业又是什么。无论是生活还是

学习工作，无论小事大事，人们都会或多或少地参考别人的选择，甚至跟随大多数人的选择作出自己的决策。这些有意识的参考或是模仿、跟随，就是羊群行为。

专栏 10.1 凯恩斯的选美竞赛理论

最早将羊群行为这个概念引入金融领域的是凯恩斯，在其著作《就业、利息和货币通论》中，他用了一个浅显易懂的例子阐述了羊群行为及其内在动因：职业投资就像是参与报纸上的选美竞赛。读者需要在报纸公布的100张照片中选出最漂亮的6位，如果读者选出的6张照片与统计票数最高的6张完全一致，则可获得奖励。凯恩斯指出，每一个读者往往会猜测公众的普遍心理，选择被大家公认最漂亮的6张照片，而并不是自己心中最漂亮的那6张。同理，在投资股票这个斗智的游戏中，也其实只需预测几个月后由成规所决定的市场估价，而非估计这项投资在几年后的前景并计算收益。

资料来源：KEYNES J M. *The general theory of employment, interest and money*[M]. Harcourt, Brace, 1936.

金融市场中羊群行为的含义相当丰富，学者们在长期的研究中提出了诸多不同的见解。Scharfstein 和 Stein(1990)认为，羊群行为是投资者简单地模仿其他人的决策，而忽略了自己的私有信息。虽然从社会层面看，这种行为是无效率的，但这对在乎自身名誉的投资者而言却是一个理性的选择。Lakonishok、Shleifer 和 Vishny(1992)认为，羊群行为是指投资者在同一时间段内，与其他投资者一样购买或出售相同的股票。这其实是一个相对宽泛的定义，将投资者所有趋同的决策均定义为羊群行为。但是，试想某家上市公司的经营业绩出现预料之外的下滑，导致投资者纷纷卖出该股票。这一行为仅仅是在共同的信息基础上作出了一致的判断，而并非是投资者间的相互模仿。Bikhchandani 和 Sharma (2000)缩小了这一定义的范围：如果一个投资者并不清楚他人的行为，就单纯根据私人信息选择投资(或不投资)，但是在他发现其他投资者没有投资(或投资)后，就放弃自己原先的决定而跟从其他投资者。该定义强调了羊群行为是投资者一种有意识的模仿，而非相似信息或处境一致导致的行为一致。

根据以上给出的定义，可以从以下两个层面阐述金融市场中羊群行为的含义：第一个层面是狭义的羊群行为，指投资者有意模仿、跟从他人进行决策，而忽略了自有信息的行为，在 Çelen 和 Kariv(2004)的研究中，他们也将其定义为信息瀑布(information cascade)；第二个层面则是广义的羊群行为，除包含狭义羊群行为所指情形外，还包含投资者那些相同处境下采取相似行为的情形。

10.2 羊群行为的心理学解释

通过前面的学习可知道，参与经济活动的个体绝非是纯粹的理性人，而是具有多种多样复杂的心理活动，这些心理活动本身可能就构成羊群行为的最深层次的根基。因此，本节主要从社会心理学的角度来讨论羊群行为。

1. 从众心理与羊群行为

阿伦森在《社会性动物》(2001)一书中对从众(conformity)作出如下定义：由一个人或

一个团体的真实的或臆想的压力所引起的人的行为或观点的变化。他解释了从众的两个原因:①别人的行为使他认识到自己最初的判断是错误的;②他可能希望在团体中免受惩罚(如不被别人排挤和嘲笑),或者希望得到奖赏(如得到别人的喜爱或被别人接受)。

社会心理学家沙赫特(1951)曾指出,群体成员会更为喜欢与群体保持一致的成员,对于偏离群体的个人往往会厌恶、拒绝和制裁。因此,从众其实是在群体一致性的压力下,个体采取的试图解除自身与群体之间的冲突、增强安全感的一种手段。实际存在的或头脑中想象到的压力,会促使个体产生符合社会或团体要求的行为与态度。个体不仅在行动上表现出来,而且在信念上改变了原来的观点,放弃了原有的意见,从而产生从众行为。

而对应到金融市场中,不确定的信息环境以及股票价格随机游走的本质导致几乎没有人能够准确地判断市场未来的走势,各种观点众说纷纭。这种情形下,人们更容易受到从众心理的影响,保持自身观点或决策与大多数人一致。如果群体中每个个体都存在从众心理,相互模仿、相互影响,就导致整个群体表现出羊群行为。

从本质来讲,从众行为只是个体的一种心理反应外化而成的适应性行为,是由个体本身的特点和愿望引起的。一般情况下,个体经验不足、自信度低,都容易产生从众行为,并且,从众行为结果一般是可以预见的,一般不会产生恶性后果,这点也是从众行为与下文将要提到的群体心理及行为的区别所在。

专栏 10.2 餐厅点菜时的羊群行为

曾有人做过这样一个实验,观察人们在餐厅点菜时的行为是否受到推荐菜的影响。

实验在一个中等连锁餐厅——眉州东坡进行。这是一个在北京有 13 家分店的连锁餐厅,每一家分店有将近 50 张桌子,包括 60 道菜的相同菜单,不同分店之间的菜系偏爱度有差别,中等规模。

实验将餐厅提供的菜单分为以下三种类型。

(1) 控制组:菜单上没有特别标注。

(2) 排列组:对点单量最多的前五位有一个排序。

(3) 突出组:随机放置了 5 个菜系,3 个 TOP5,两个随机菜系(这些都没有告诉就餐者)。

实验实施时,工作人员随机选择 5 家分店,在里面随机放置控制组和排列组;随机选择 4 家分店,放置控制组和突出组。

实验结果显示:当畅销榜对消费者有暗示时,对排行前五的菜的需求上升 13%~18%。证据显示观察学习在不常光顾的消费者中有更大的影响力。听从其他消费者的选择信息而非样品菜时,消费者的进餐体验水准也被提升了。

资料来源:CAI H,CHEN Y,FANG H. *Observational learning:evidence from a randomized natural field experiment*[J]. *American economic review*,2009,99(3):864-882.

2. 群体心理与羊群行为

人总是生活在一定的社会环境中,依从于各种各样的社会关系,参加一定的群体生活。这种群体既包括客观群体——人群在某一时间和空间上的集合体,也包括心理群体——具有相同心理活动特征的人群的集合体。

法国社会心理学家勒庞(Le Bon)(1895)在《乌合之众——大众心理研究》一书中首先提出了“心理群体”这一名词,为研究群体心理学奠定了基础。

勒庞认为群体心理是指个体处于心理群体中时所产生的心理状态和心理倾向。这主要包括两层意思:一方面,这种心理状态与个体独处时的心理状态完全不同,是在群体中才会具有的一种心理倾向,是在群体行为中表现的一种群体心态;另一方面,只有个体具有了相同的心理倾向才能聚集成群体,形成群体心理。群体的聚集过程中个体所形成的这种心理状态和心理倾向在平常是被理性所压抑的,处于潜在的状态,而当心理群体形成之后,个体就会冲破道德、伦理的约束,而表现出一种原始的冲动,外化表现出共同的行为。

勒庞指出,心理群体的形成过程主要经历了以下三个阶段。

(1) 情绪激发——人群中的个体由某一情绪所激发并最终被情绪所控制。

(2) 情绪传递——人群中的个体情绪由激发进而相互传递(相互感染并最终控制整个群体)。

(3) 建议接受——当人群被某一情绪控制后,人群便易受到外部建议的控制并受其指挥而参与某一激烈的行动。

当心理群体形成后,它作为一个整体的心理状态具有以下三项基本特征。

(1) 冲动性。冲动性指心理群体的集体行为可以在外界某一强刺激因素的激发下迅速启动,并且这种行动也可以在另一强刺激因素的作用下迅速向相反方向逆转。因此,心理群体情绪的高度不稳定性必然导致其行动方向的极度不确定性。

(2) 服从性。服从性指心理群体具有自愿服从外来指挥的心理特征。由于在心理群体中,所有成员的情感与观念都被统一到一个方向,理性化的个性成分也都被消除,因此他们具有自愿服从他人意愿的强烈倾向。

(3) 极端性。极端性指心理群体的实际行动常常超过必要的限度,因此往往导致过激的后果。

当人群形成一个心理群体时,该群体作为一个整体在决策和行为时所表现出来的智能水平将远远低于该群体成员在作为单个个体决策和行动时所能表现出来的智能水平。这就是著名的勒庞“心理群体智能低下理论”。

心理群体形成的这三个阶段,展现了群体性事件羊群行为形成的过程中,群体中的个体的情绪是如何相互影响,并最终形成了一致的心理状态,而这种一致的心理状态,或者说群体心理的产生,可以看作羊群行为形成的一种根源。根据勒庞的“心理群体智能低下理论”,心理群体一旦形成,群体行为中的每一个个体的智能水平将远低于个体正常的智能水平,其理性水平和分辨能力也处于最低状态,导致群体行为能够最大限度地突破人类理性的限制。这也会导致群体的心理状态和行为状态很容易失控,行为的结果往往很难预料,而这种不确定性与群体的规模往往存在着正向的联系。

专栏 10.3　美国几次重大事件中群体认知的偏差

我们都听过这样一句谚语,“三个臭皮匠,顶个诸葛亮”。这句谚语旨在告诉我们,人多智慧多,解决难题时如果能集合大家的智慧,更有可能商量出一个好办法来。但勒庞的

“心理群体智能低下理论”却告诉我们一个不同的结论——如果部分个体拥有强烈的情绪,并与其他个体相互影响,最终产生的集体决策有可能就不再理性。

当金融市场被过度自信或者强烈恐慌等情绪所笼罩时,群体认知往往与实际情况存在巨大偏差,近百年来美国所发生的多次重大事件都证明了这一结论,如表 10.1 所示。

表 10.1 近百年来美国所发生的多次重大事件

时间与事件	大多数人的一致看法	实际情况
1929 年	股市已经站稳了“永久的价格平台”,牛市将长期持续	出现了美国历史上最严重的经济衰退和熊市
1933 年美国宣布放弃金本位制度	将出现通货膨胀和股市熊市	未出现通货膨胀,股市进入牛市
1955 年朝鲜战争结束	将出现经济衰退和股市熊市	未出现衰退,股市进入牛市
2000 年	新经济时代来临,纳斯达克指数还将继续攀升	经济硬着陆,股指全线下跌
2008 年	衍生工具等金融创新虽有风险但风险可控,华尔街巨头有政府兜底“大而不倒”	金融过度创新隐藏巨大风险,大型金融机构破产引发多米诺效应,金融危机爆发

10.3 羊群行为的理论模型

1. 信息流羊群行为

Bikhchandni、Hirshleifer 和 Welch(以下简称“BHW”)在 1992 年提出了信息流(information cadscade)羊群行为理论:在不确定的市场中,每个投资者面临相似的投资决策,且均拥有不完美的私人信息,投资者可以观察到他人的决策并推测他人的私人信息,但不能直接观察他人的私人信息。在这种框架下,起初的一些随机事件和少数人的行为选择往往会决定后来大部分人的行为模式,人们会忽视自己拥有的私人信息,作出和前人一致的选择,这就导致了基于信息流的羊群行为。

这一模型假设:个体按照外生给定的顺序来进行决策,决策的内容为是否以 1/2 的价格买进一种资产。资产的价值是 0 或者 1,概率均为 1/2。每个投资者都会观察到一个关于资产价值的不完美的私人信号,信号为 H 或 L。观察到信号 H 意味着资产真实价值为 1 的概率为 $p(p>1/2)$,资产真实价值为 0 的概率则为 $1-p$。观察到信号 L 的意义相反,意味着资产真实价值为 0 的概率为 p,为 1 的概率为 $1-p$。每个个体可以看到在他之前所有个体的决策,并依此推断他们的私人信息,如图 10.2 所示。

可以设想:若第一个投资者得到的私人信号为 H,他应当选择买进,而若信号为 L,他应当选择不买入。若第二个投资者发现第一个投资者选择买进资产,并且他得到的私人信号为 H,则第二个投资者会同样选择买进资产。若此时他的私人信号为 L,那么在他看来,资产的期望价值应当是 1/2,这也就意味着他选择买进的概率为 1/2。对于第三个投资

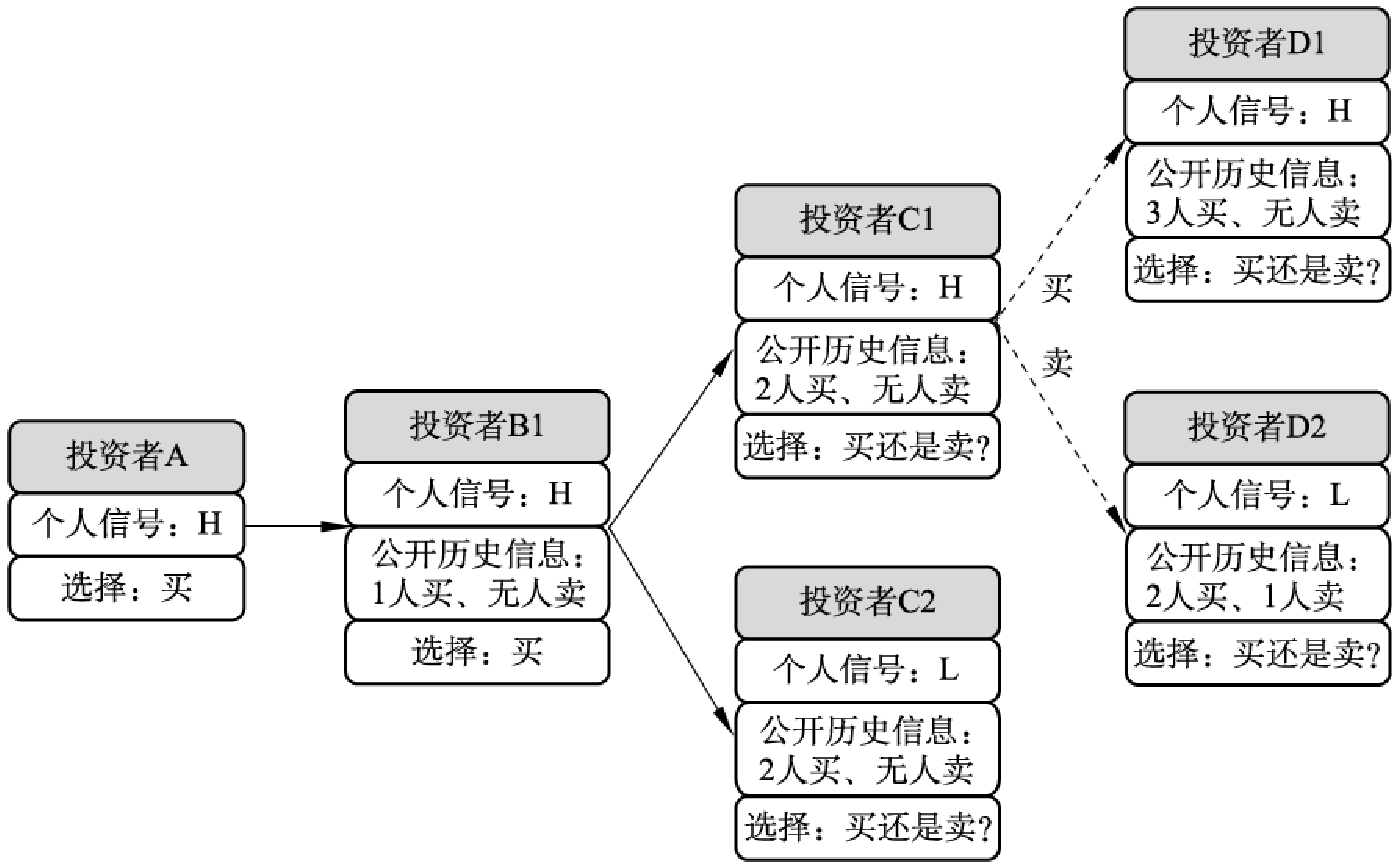


图 10.2　信息流模型结构示意图

者，他将面临三种情况：①在他之前的两名投资者均买入（此时即使他得到 L 信号，他也应当选择买入，这样一来，便形成了一个“买入”的决策序列）；②两名投资者均不买入（此时即使他得到 H 信号，他也应当选择不买入，这样一来，便形成了一个“不买入”的决策序列）；③一名选择买入，另一名选择不买入。在最后一种情况下，第三名投资者面对的情况和第一名投资者相仿：依据前人行为，他对资产的价值期望为 1/2，所以他个人得到的信号将成为他决策的依据。在这个模型中，第三名投资人面对前两名投资人“买入”、“买入”的公开信号时，无论他自己的私人信号是 H，还是 L，他的选择都应该是“买入”。于是，信息流形成。同理，如果前两人选择卖出，第三人也应该选择卖出。对于第四名、第五名甚至之后更多的投资者，都可以按照上面的逻辑进行类推。

从上面的分析中可以看出，第一个投资者和第二个投资者的行为具有信息含量，可以为后继投资者提供参考，从而增加了公共信息量。而一旦形成信息流，后继投资者依照决策序列进行决策，他们的行为便没有任何信息含量，其私人信息也无法转化为公共信息，最终导致公共信息的贫乏和市场无效率。

同时，BHW（1992）的研究也指出，这种信息流羊群行为模式是相当脆弱的，一旦有一点新的信息产生，这种行为就有可能发生改变。这是因为，当投资者认识到自己身处某种信息流中时，他也知道这种信息流所依赖的信息相对于投资者的私人信息来说是相当贫乏的。

2．声誉羊群行为

Scharfstein 和 Stein（1990）从委托代理的角度出发，构建了声誉羊群模型。两个风险中立的代理人先后投资于两个相同的投资项目。代理人分别为聪明的和愚笨的两个类型，聪明的代理人有更高的概率得到正确的信号，而愚笨的代理人只能得到噪声信号。由于代理人本身并不清楚自己是聪明的还是愚笨的，为了不让雇主认为自己是愚笨的代理

人，他们都会选择跟随对方的行为，于是就产生了羊群效应。他们在研究中首次提出了“共同承担责备效应”(blame sharing effect)这一观点。他们指出，如果某决策者作出了有悖于大多数人的决策，一旦他失败了，就会被雇主认为能力有限，并因此而受到责备(可能是降薪或者被解雇)。但是如果他的行为与大多数人一致，即使失败了，他的雇主也会考虑到其他人同样失败了而不过分责备他，同时他自己也不会过分内疚。因此，决策者具有与别人趋同的决策倾向，以推卸自己因错误决策而必须承担的责任。虽然这种行为从雇主或社会角度看是无效率的，但从决策者自身来看确实符合自己的最大化利益，是个人理性的选择。

Graham(1999)继承并发展了该理论，他提出，投资经理的雇主和其他人一般会通过与其他投资经理的业绩进行比较来判断该投资经理的水平。雇主的判断将影响经理人在劳动市场上取得的声誉，进而影响其当前以及下一份工作能够获得的报酬。投资经理人的任何投资决策都将面临声誉成本，而模仿其他经理的投资决策将最大限度地降低声誉成本。因此，经理人倾向于模仿他人的决策。

专栏 10.4 股票分析师荐股的羊群行为

分析师对走势的判断往往影响到自身职业发展，Hong，Kubik 和 Solomon(2000)曾指出，分析师出于对声誉和长期职业生涯的考虑，往往倾向于和其他分析师保持一致，羊群行为程度较投资者更高。他们还指出，相比起经验不够丰富的分析师，有经验的分析师往往更倾向于提供冒险的预测而且更不容易被解雇。

的确，当前国内分析师的研究报告几乎都是无偿向大众公开，当市场情绪浓烈到一定程度时，任何有违公众认知的观点都无疑会遭到强烈的质疑甚至抨击。于是更多的分析师选择了顺从多数，市场判断的信息含量和准确度也就大打折扣。表 10.2 列示了 2008 年多家知名券商研究报告的标题和大致内容。

表 10.2 2008 年多家知名券商研究报告的标题和大致内容

报告名称	大致内容
《高估值将持续》	“中国 A 股市场的高估值状况将在 2008 年持续” “在国内流动性充裕的情况下，A 股的估值仍有进一步扩张的可能性”
《A 股总体处于震荡整理》	“沪深 300 指数在 4 500～7 000 点，区间内波动概率较大，市盈率波动区间为 24～37 倍”
《2008 年 A 股市场投资策略》	2008 年牛市的承前启后执念，在充分消化了新的宏观信息后，未来仍是大牛市，但 2008 年收益的波动性大幅提高，乐观的话可以看到 8 000 点，而下跌空间有限
《牛市将有结构性变化》	中国股市已经不再是“初级阶段”，但明年业绩、两税合并等因素还在支撑牛市的运行，但波动幅度会明显加大
《踏遍青山人未老》	2008 年 A 股市场依然维持牛市格局，保守预计 2008 年沪深 300 指数为 7 200 点
《光荣与梦想》	从长期和动态来看，目前的高估值有其合理性，上证综指全年在 4 500～6 000 点内波动

3. 基于薪酬条款的羊群行为

对于业绩激励与羊群行为的研究始于 Brennan(1993)和 Roll(1992)等人。他们提出如果基金经理的报酬取决于自身业绩与他人业绩的比较，这将会诱使基金经理作出无效率的投资决策，甚至导致羊群效应。Maug 和 Naik(1996)所提出的基于薪酬条款的羊群行为模型(compensation-based herding)则进一步解释了上述观点。模型假设一个风险厌恶的经理人，他的薪酬取决于相对于基准经理人的表现：当表现优于基准时，薪酬增加；当表现比基准差时，薪酬下降。所有人包括基准经理人都拥有关于股票收益的不完美的私人信息。假定基准经理人先作决定，其他经理人在观察到基准经理人的决策结果后再作决定。因而，在相对业绩激励机制下，经理人有动力去模仿基准经理人，因为他们的最优投资组合就是与基准投资组合保持一致。此后，Admati 和 Pfleiderer(1997)进行了拓展，分析了多重风险资产的模型，发现与基准挂钩的薪酬合约下，经理人的决策的确会导致羊群行为，而且羊群行为并不能有效克服道德风险和逆向选择。

专栏 10.5　基金绩效排名与羊群行为

基金绩效排名对基金经理人的声誉和薪酬都会产生影响，面对排名压力，基金经理人在投资决策上会更多地参考好的基金，以提高自己的排名吗?

路磊等(2014)对我国证券市场基金的羊群行为进行了研究，他们发现当基金排名上升时，基金经理人会更多地参考好的基金，以维持自己的排名；而当基金排名下降时，他们会减少对好基金的参考，因为模仿其他的基金进行投资并不能使自己的排名超过他人，为了赶超其他基金，必须改变投资策略。此外，他们的研究还发现，相对于中资基金，合资基金对排名变化的敏感性较低，合资基金公司的羊群效应变化与其排名变化没有显著的关系。

资料来源：路磊，黄京志，吴博. 基金排名变化和羊群效应变化[J]. 金融研究，2014(9)：177-191.

10.4 羊群行为的实证检验

目前，针对羊群行为的检验方法多为实证检验，即利用统计学方法，研究证券市场上投资者的投资决策是否存在趋同的现象。

1. LSV 法

LSV 法是由 Lakonishok，Shleifer 和 Vishny 在 1992 年提出的，他们将羊群行为定义为投资者同时买卖某些特定股票的平均趋势。因此，他们提出的这一指标实际上测度的是交易者对于某种特定股票交易模式的共同趋势以及趋同的程度。

以 $B(i,t)$，$S(i,t)$表示在 t 时间段内对股票 i 买、卖的投资者数量，$H(i,t)$为羊群行为的测度值，则 $H(i,t)$可表示为

$$H(i,t) = |p(i,t) - p(t)| - AF(i,t)$$

其中，$p(i,t) = B(i,t)/[B(i,t) + S(i,t)]$；$p(t)$为 $p(i,t)$对股票 i 的均值。调整因子 $AF(i,t) = E[|p(i,t) - p(t)|]$。如果 $H(i,t)$显著不为 0，则表示羊群行为存在。

LSV 法是最早提出的关于检验羊群行为的方法，但这种方法也存在以下两点缺陷：

①仅用买卖双方数量来衡量羊群行为，忽略了交易量的问题。②仅能判断某只股票的买卖是否有羊群效应，但没有考虑是哪些因素导致趋同行为的产生。

2. PCM 法

Grinblatt，Titman 和 Wermers(1995)运用投资组合变动度量(PCM)方法，来测量管理投资组合交易在方向和力度上的羊群行为，适用于判断各类投资基金间是否存在羊群行为。

对于投资组合 I 和 J，时间延迟为 τ 的横截面，羊群行为度量指标为

$$\hat{\rho}_{t,\tau}^{\mathrm{I,J}} = \frac{\left(\frac{1}{N_t}\right)\sum_{n=1}^{N_t}(\Delta\tilde{\omega}_{n,t}^{\mathrm{I}})(\Delta\tilde{\omega}_{n,t-\tau}^{\mathrm{J}})}{\hat{s}^{\mathrm{I,J}}(\tau)}$$

其中，$\Delta\tilde{\omega}_{n,t}^{\mathrm{I}}$ 为时间间隔 $[t-1,t]$ 内投资组合 I 中股票 n 的持有比重变动；$\Delta\tilde{\omega}_{n,t-\tau}^{\mathrm{J}}$ 为 $[t-\tau-1,t-\tau]$ 内投资组合 J 中股票 n 的持有比重变动；N_t 为组合 I 和组合 J 在期间 $[t-1,t]$ 中的可交易证券集的交集。

PCM 法也有一些不足，主要是：第一，PCM 法用股票变动比重来衡量羊群行为，由此产生了另一个偏差，即资金量越大的基金管理人被赋予的比重越大；第二，当股票价格变动时，即使没有任何买卖行为，股票的市值比重也会变动，因此 PCM 法可能衡量出虚假羊群行为；第三，没有确认在 PCM 法中用以计算股票比重的净资产值的方法。

3. CSAD 方法

这一方法是由 Chang，Cheng 和 Khorana(2000)提出的，是目前较为通用的方法。CASD(cross-sectional absolute deviation of returns，收益率的横截面绝对偏差)指的是在某一时间横截面上，市场上各只股票收益率与市场组合收益率的绝对差。

$$\mathrm{CSAD}_t = \frac{1}{N}\sum_{i=1}^{N}|R_{i,t} - R_{m,t}|$$

其中，N 为投资组合的股票数量；$R_{i,t}$ 为股票 i 在交易日 t 的收益率；$R_{m,t}$ 为市场组合收益率。用这一指标来衡量市场整体的羊群行为的原理在于，当市场上多数投资者对于许多股票都表现出强烈的羊群行为时，股票的收益率都会出现趋同。

4. 羊群行为在证券市场中的检验

Lakonishok，Shleifer 和 Vishny 在 1992 年提出了羊群行为检验的 LSV 方法，并利用该方法对美国证券市场进行了实证检验。他们以 1985—1989 年美国的 769 只股票为样本，进行了 LSV 检验，得出的 H 检验值等于 2.7，并由此认为这些基金不存在显著的羊群行为。同时，他们还发现，发生在小公司股票上的羊群行为比大公司股票更为显著。Grinblatt，Titman 和 Wermers(1995)应用 LSV 方法，检验了 1974 年年末至 1984 年年末美国 274 家基金，得到了同样的结果——这些基金在投资中并无明显的羊群行为。1995 年，Wermers 提出了 PCM 方法，并利用该方法进行检验，却发现这些基金在投资时存在显著的羊群行为。而 Wermers(1999)对之前采用的方法进行了改进，将分析的股票进行筛选，只留下了至少有 5 家机构活跃交易的那部分股票。他研究了 1975 年至 1994 年美国市场上所有基金的交易行为，结果发现羊群行为显著存在。Chang，Cheng 和 Khorana(2000)

利用其提出的 CSAD 方法分别检验了美国、日本、韩国以及我国台湾、香港地区的股票市场，得出结论：美国、我国香港地区的证券市场羊群行为不显著，日本市场存在一定的羊群行为，我国台湾地区和韩国市场存在显著的羊群行为。

国内关于羊群行为实证检验的研究比较多，结论也比较一致。大多数研究均认为国内证券市场存在比较明显的羊群行为。宋军和吴冲锋(2001)研究了 1992 年 1 月 2 日到 2000 年 12 月 31 日沪深两地所有上市公司的日收益率和月收益率数据，利用股票收益率分散度指标进行检验，结果发现我国证券市场的羊群行为程度高于美国证券市场的程度。并且，在市场收益率极低时，羊群行为程度明显高于市场收益率极高时的羊群行为程度，说明了投资者在市场整体低迷时更倾向于模仿他人的行为。孙培源和施东晖(2002)以 CAPM 模型为基础，建立了一个更灵敏的检测羊群行为的模型，发现我国股市存在一定程度的羊群行为，导致系统风险在总风险中比例较大。此外，他们的实证检验结果还表明，我国股市在市场极端波动时存在一定程度的羊群行为，特别是市场处于大涨时羊群行为更加明显。石荣丽和阳浙江(2004)借鉴了 Wermers(1994)的方法，筛选出 2001 年第一季度到 2002 年第四季度中至少有 5 家基金交易过的股票，并利用 LSV 方法进行检验。研究计算出的羊群行为度为 8.94，且统计意义上显著，说明在这些股票的交易中，羊群行为明显存在。

10.5 羊群行为的实验经济学检验

羊群行为虽然在证券市场上普遍存在，但它并不仅仅是一个经济学问题，其背后还隐含着人类心理学、行为学等诸多方面的理论。关于羊群行为的实证检验仅能得出纯粹统计意义上的结果，而不能分析人们行为背后的具体动机。近年来，随着行为金融理论和实验经济学的兴起，越来越多的学者开始从行为金融视角出发，利用实验经济学方法，对羊群行为的理论模型进行发展和完善。

Anderson 和 Holt(1997)以 BHW(1992)提出的信息流羊群模型为理论基础设计了一个摸球实验，并在弗吉尼亚大学招募学生完成了这项实验。实验流程大致如下：6 人为一个小组进行决策。实验需要用到两个罐子——罐子 A 和罐子 B。A 中有 2 个 a 球和 1 个 b 球，而 B 中有 2 个 b 球和 1 个 a 球。在每一轮决策开始前，实验助手将会投掷一个骰子来决定本轮实验要使用的罐子，若骰子点数为 1、2、3，则使用罐子 A，否则使用罐子 B。骰子投掷完毕后，罐子中的球将会被放进一个没有标记的容器中，被试并不知道本轮使用的究竟是哪一个罐子，如图 10.3 所示。

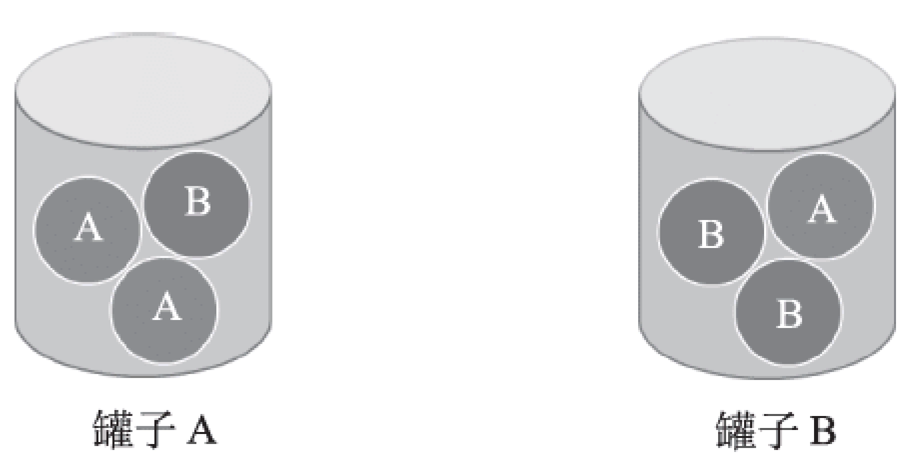


图 10.3 Anderson 和 Holt(1997)实验示意图

在每一轮中，实验助手按照随机给定的顺序依次为每名被试抽一次球，每次抽取的结果只有该名被试一人可以得知。第一名被试获得自己的抽取结果后，需要猜测本轮使用的是哪一只罐子，并将自己的猜测结果报告给实验助手，实验助手将把他的决定公布给其他的被试，以此类推，直到第六个被试作出他的猜测。然后，实验助手将公布本轮使用的罐子信息，猜对了的被试可以得到 2 美元的奖励，而猜错了的被试则得不到任何奖励。实验按照这样的规则共进行 15 轮。

根据 BHW(1992)的理论，当第一个和第二个被试作出了相同的猜测，后续所有被试都会跟随他们。也就是说无论被试得到什么样的私人信号，他们都会作出和第一、第二个被试一样的选择，这就形成了信息流。

实验结果显示，在 12 场共 122 轮成功进行的实验中，有 87 轮实验形成了信息流。在激励机制比较简单的实验环境下，信息流比较容易形成。在几轮实验中，最初的错误信号启动了一条错误的信息流，而且在之后的决策中，也并没有相反的决策来终止这一信息流，最终导致所有被试的决策错误。这也说明了羊群行为会导致公共信息量无法增加、决策无效率的问题。

Anderson 和 Holt(1997)的实验为 BHW(1992)的信息流羊群模型提供了有力的证明。但在他们之后，一些学者却提出了质疑，学者们在这一实验的基础上修改了部分细节，得出了与原实验不一致的结果。Huck 和 Oechssler(2000)对实验被试进行了严格的筛选，保证他们能够准确理解贝叶斯规则。他们挑选了经济学、商务管理专业的学生，并在他们的微观经济学考试试卷中测试了他们对贝叶斯规则的掌握程度。同时他们将实验激励设置为期末考试的额外加分，从而保证了被试们决策目标的一致性。另外，他们还在实验结束后，要求被试解释自己作出选择的依据。他们的实验结果显示，在一半左右的交易中，被试的行为服从了贝叶斯规则，但极少有被试能够准确地解释自己的行为为什么是服从贝叶斯规则的。这意味着，被试的选择只是恰巧符合了贝叶斯规则，而并非 Anderson 和 Holt(1997)所认为的“基于贝叶斯规则的理性选择”。更多的被试选择用“跟随自己的信号”这一规则来解释自己的行为。

此外，还有许多学者认为原有模型中的市场环境比较简单，模型假设与现实市场存在较大区别。因而，他们对模型中原有的信息结构、价格机制等方面进行了修改，试图对原有的理论作出补充。Noth 和 Weber(2003)增加了私人信息的种类，引入两种不同质量的信息。他们通过电脑程序进行实验，增加了实验的轮数，以期观察被试在重复实验中的学习效应。实验结果显示，被试在整合私人信息和公共信息时，往往会出现过度自信的现象，即过分相信并依赖于自己的信息。在前面决策者的选择不统一时，过度自信的现象更加明显。Cipriani 和 Guarino(2005)并没有采用抽球的实验机制，而是直接采用了 BHW(1992)的理论模型中的情景，将实验设定为对某种资产进行投资。而且，他们放宽了“所有投资者均面临相同价格下的投资机会”这一假设，引入了价格浮动的机制，将价格设定为依据前人的交易信息计算出的资产期望价值。实验结果显示，在价格浮动机制下，羊群行为发生的概率大大降低。但他们同时也发现，一些被试会采取高风险的投资行为——同时违背自己的信号以及大众的选择，以期获得更高的收益，这类行为被称为逆向

行为(contrarian behavior)。

课程实验指引[①]

1. 实验背景

本实验将建立一个证券市场,每个参与人作为一名投资者,在市场中进行交易。每轮实验开始前将赋予你一定数量的账面现金作为你拥有的初始资产。你的资产会由于你的投资决策相应增减,这与你最终得到的收益挂钩。具体来说,实验将按照如下方式进行。

2. 实验周期

本次实验共13轮,也就是13个交易周期。前3个交易周期是练习期,用来让大家熟悉交易规则以及操作系统,这几个周期的收益不会计入你最终的报酬。实验从第4个交易周期正式开始,第4~13个交易周期与你最终得到的收益挂钩。

3. 交易资产

每位投资者在每一轮交易开始前,都拥有账面现金资产100点。实验中,我们要交易的资产是1个单位的某种股票。在这一轮结束时,每单位股票的赎回价值有可能是100点,也有可能是0点。两种情况的概率均为50%。除了赎回价值,该股票无任何价值。股票赎回价值究竟是多少是由电脑系统决定的,只有在每一轮结束的时候投资者才会知道。

4. 交易顺序

每个市场(每个实验小组)由6名投资者(由各位实验被试扮演)和1个做市商(由电脑系统扮演)组成。实验开始后,在每一轮中,各位投资者按顺序依次作出交易选择。交易的顺序是由系统随机决定的。

5. 交易机制

本市场采用做市商交易机制,所有交易者与做市商进行交易。你可以选择买进1个单位或者卖空1个单位的股票(必须在买进或者卖空中选择,不可以不进行任何交易)。每轮实验中,买进和卖空的市场价格均恒定等于50点。买进股票指的是以市场价格 P 买进股票,本轮结束时赎回,得到股票赎回价值 V。卖空指的是向做市商借入股票并立即以市场价格 P 卖出,在本轮结束后按赎回价值 V 买入股票,还给做市商。买进和卖空是方向恰好相反的两种操作。

简单来说,如果买进,你的成本是 P,最终收益 $=V-P$;如果卖空,你的成本是 V,最终收益是 $P-V$。当你觉得赎回价值高而现在价格低,那么就买进;相反当你觉得赎回价值低而现在价格高,你应该卖空。

在你每一次进行选择之前,你会看到在你之前的几位投资者都做出了怎样的选择。并且,你将得到一个系统给出的信息,告诉你股票在本轮结束时的赎回价值,这个信息只

---

① 本部分主要参考 Anderson 和 Holt(1997)以及 Cipriani 和 Guarino(2005)的设计。实验目的是检验个体的决策行为是否会形成信息流。

有你自己知道。但需要注意,这个信息的准确度只有70%。

6. 实验报酬计算

实验报酬按照如下的方法计算。

首先,计算正式实验(4～13轮,共10轮)中,每轮交易结束后各投资者的净资产:每位投资者管理的初始资产都是100点。在之后的交易中,若你选择买进股票,净资产=100点-市场价格+股票赎回价值。若你选择卖空,净资产=100点+市场价格-股票赎回价值。

下面举两个简单的例子来说明净资产的计算方法。

如果你买入股票,价格为50点。本轮结束时,股票赎回价值为100点,那么你的净资产就变为:100点-50点+100点=150(点)。

如果你卖出股票,价格为50点。本轮结束时,股票赎回价值为100点。那么你的净资产就变为:100点+50点-100点=50(点)。

最后随机抽取一轮,得到该轮交易结束后你的净资产,按照一定比例,将实验中的点数化为人民币,计算得到你在该轮实验中的收益,这也是你在本实验中得到的最终报酬。

本章小结

(1) 金融市场中的羊群行为有两个层面的含义:第一个层面是狭义的羊群行为,指投资者有意模仿、跟从他人进行决策,而忽略了自有信息的行为;第二个层面则是广义的羊群行为,除包含狭义羊群行为所指情形外,还包含投资者那些相同处境下采取相似行为的情形。

(2) 信息流羊群行为是指在不确定的市场中,每个投资者面临相似的投资决策,且均拥有不完美的私人信息,投资者可以观察到他人的决策并推测他人的私人信息,但不能直接观察他人的私人信息。在这种框架下,起初的一些随机事件和少数人的行为选择往往会决定后来大部分人的行为模式,这就导致了羊群行为。

(3) 声誉羊群行为是指在委托代理关系中,由于雇主缺乏有效的手段评判代理人的业绩,代理人为了不让雇主认为自己是愚笨的,往往都会选择跟随对方的行为,于是就产生了羊群效应。

(4) 共同承担责备效应是指如果某决策者作出了有悖于大多数人的决策,一旦他失败,就会被雇主认为能力有限,并因此而受到责备(可能是降薪或者被解雇)。因此,决策者具有与别人趋同的决策倾向,以推卸自己因错误决策而必须承担的责任,同时减少自己的内疚。

(5) 基于薪酬条款的羊群行为是指如果经理人的报酬取决于自身业绩与他人业绩的比较,经理人有动力去模仿代表业绩基准的那位经理人,这将会诱使基金经理作出无效率的投资决策,甚至导致羊群效应。

思考题

1. 从众行为和群体行为有什么区别？试着举出生活中的例子加以说明。

2. 羊群行为是否都是不理性的？是否都会造成不利的结果？请举例说明。

3. 金融市场中存在的羊群行为加剧了市场的波动。你认为可以从哪些方面入手来减弱羊群行为，从而维护市场稳定？

第 11 章

金融市场价格泡沫

引导案例：日本经济泡沫

1986 年 12 月到 1991 年 2 月是日本在第二次世界大战后，仅次于 20 世纪 60 年代后期经济高速发展的第二次大发展时期。但是好景不长，这场受大量投机活动支撑的经济浪潮如昙花一现，在 90 年代初泡沫破裂，日本经济出现大倒退，此后进入平成大萧条时期。

1. 泡沫的产生

20 世纪 60 年代中期，日本为了避免外国企业对本国企业的恶意兼并，实施了稳定股东计划，企业与银行之间、企业与企业之间相互持股，形成你中有我、我中有你的相互约束机制。因此，各法人很少买出股权，导致股票市场的供给短缺、需求上升，股票价格也随之一路攀升，形成“股票不倒”的神话。

1985 年 9 月 22 日，世界五大经济强国（美国、日本、德国、英国和法国）在纽约广场饭店达成“广场协议”。当时美元汇率过高而造成大量贸易赤字，为此陷入困境的美国与其他四国发表共同声明，宣布介入汇率市场。此后，日元迅速升值。当时日本政府为了补贴因为日元升值而受到打击的出口产业，在 1986 年至 1987 年连续 5 次下调利率，并多次入市干预，于是产生了过剩的流通资金。宽流动性在很大程度上助推了股市的泡沫。

与此同时，从 20 世纪 70 年代后期开始，日本银行面向优质制造业公司的融资业务普遍不理想，于是开始倾向于向不动产、零售业、个人住宅等融资。20 世纪 80 年代，日本为使国营铁路私有化，以每股账面 5 万日元，也就是 20 倍的溢价向私人抛售铁路公司的股票。东京股市立即掀起了一场投机热潮。

2. 泡沫的推进

从 1985 年年底到 1990 年年初，短短 5 年多时间，日本城市地价增长了 200%。1990 年，仅东京的地价就相当于美国全国的土地价格，而美国的面积则是东京的 1.5 万倍。无

论是实体企业还是金融机构，都不计成本将资金投入土地市场。铁路公司的股价也由每股 5 万日元迅速飞涨到每股 300 万日元。随之而来的"圈地运动"，使得地价越炒越高，东京市区每 3.3 平方米竟高达 1 亿日元，成为当时世界上最昂贵的"黄金地产"。

房地产投机泡沫和股市投机泡沫相互交叉、彼此推进。到 20 世纪 80 年代中后期，日经 225 指数股票平均价格已飙升到 4 万日元，不少人在短期内就从中获得了百分之百以上的暴利。

3. 泡沫的破灭

到 1989 年中期，日本货币当局一方面为了防止普遍物价上涨，另一方面意识到资产阶级价格膨胀的严重性，于 5 月 31 日将官方贴现率提高到 3.25%，又于年末进一步提高到 4.25%，同时大藏省也采取了多项措施限制土地投机。1990 年 8 月，以"海湾危机"为契机，为防止石油价格上涨导致通货膨胀，日本中央银行采取了预防性紧缩政策，将官方贴现率提高到 6.0%。1989 年 12 月 29 日，日经平均股价最高达到 38 915.87 点，此后开始下跌，如图 11.1 所示。土地价格也在 1991 年前后开始下跌，泡沫经济开始正式破裂。到了 1992 年 3 月，日经平均股价跌破 2 万点，仅达到 1989 年最高点的一半，8 月，进一步下跌到 14 000 点左右。大量账面资产在短短的一两年间化为乌有。金融体系受到重创，并进而影响到实体经济，成为日本经济 20 世纪 90 年代持续衰退的一个重要原因。

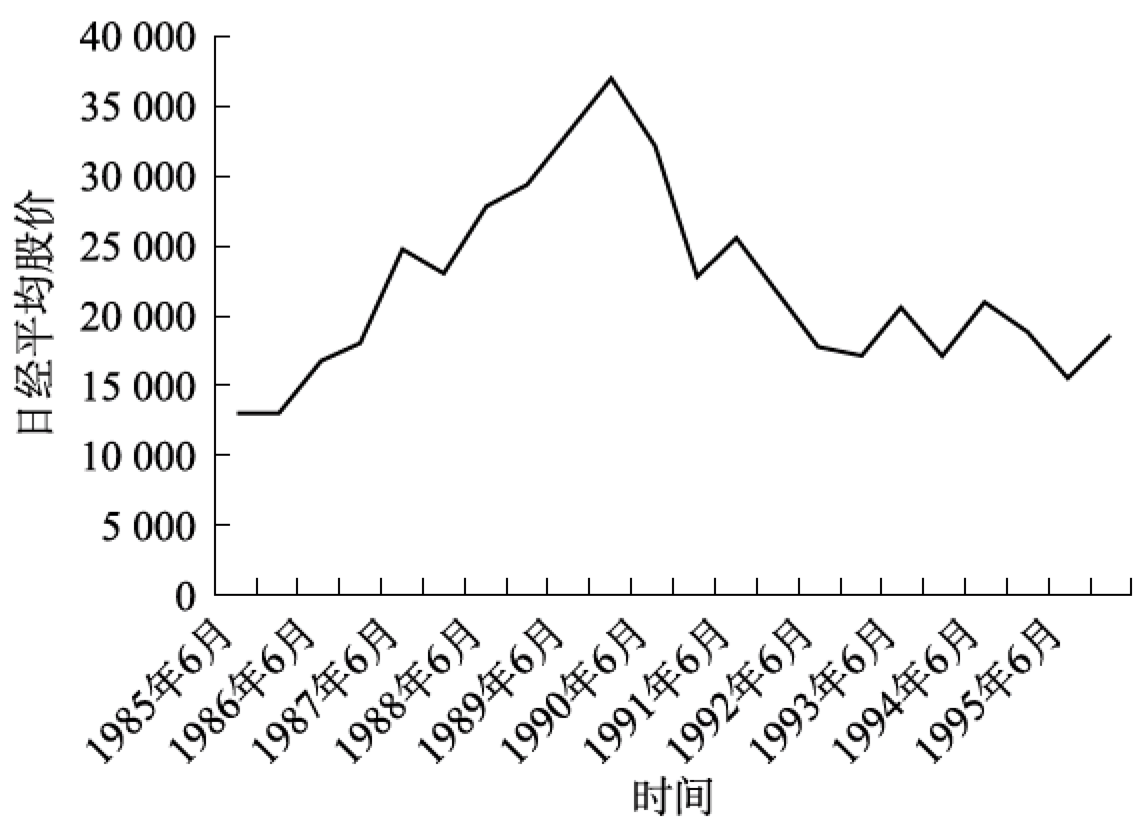


图 11.1　1985—1995 年日经平均股价

自此 20 年来，日本六大主要城市住宅用地价格跌幅达到 65%。随着房价暴跌，国民财富持续缩水，损失高达 1 500 万亿日元，相当于日本 3 年的 GDP 总和。日本经济更是萎靡不振，20 年间，日本年均 GDP 增速只有 0.75%。

日本房地产经济与股市完整演绎了泡沫从生成、推演到破灭的全过程。投机促成了泡沫的生成，各类投资者助推泡沫愈演愈烈，使得股票与房地产价格偏离基本价值而急剧上升，直到泡沫被市场发现，价格回落，泡沫最终破灭，引发严重金融危机。那么，金融市场泡沫究竟是什么呢？它有怎样的特征表现？它为什么会产生以及怎么去判断市场中是否有泡沫？这些问题都将在本章中得到解答。

11.1 金融市场泡沫概述

1. 金融市场泡沫的定义

泡沫(bubble),是指一种或一系列的金融资产在经历连续的涨价之后,市场价格大于实际价值的经济现象。其产生的根源是过度投资引起资产价格的过度膨胀,导致经济的虚假繁荣。

Kindleberger(1978)在《新帕尔格雷夫货币与金融学辞典》中对泡沫的定义是:“一种资产或一系列资产价格在一个连续过程中的急剧上涨,初始的价格上涨使人们产生价格会进一步上涨的预期,从而吸引新的买者——这些人一般是以买卖资产牟利的投机者,他们交易此资产的目的是通过交易来获利,而不是想使用它。”

Stiglitz(1990)在《有关泡沫的讨论》一文中指出:“如果价格今天很高的原因仅仅是因为投资者相信明天的销售价格将会更高——而基本面因素不能解释这样的高价格——那么就存在泡沫。”他同时指出,如果资产价格在没有任何新信息的情况下显著下降,这就能被定义为泡沫破灭。

亲历了日本地产与股市泡沫的三木谷良一(1998)认为,所谓投机泡沫就是资产价格严重偏离实体经济暴涨,然后暴跌这一过程。

泡沫会发生在社会经济的各个方面。不论是实物商品还是金融资产,都不可避免地受到泡沫的冲击。如陕西旬阳鸡血石 1 年价格暴涨 10 倍,普洱茶每 100 克高达 22 万元的拍卖价格,股票指数的异常大幅度上涨,都无一例外有着泡沫的影子。

2. 金融市场泡沫的特征

虽然历史上不同时期、不同地点在不同产品上出现了各式各样的市场泡沫,但是它们的产生、推进和发展都表现出相对一致的特点。

1) 价格与价值严重背离

Garber(1990)将资产价格泡沫定义为“不能被基本面因素所解释的价格运动”。换言之,泡沫是金融资产的持续上涨,使得价格严重偏离了实际价值。这是金融市场泡沫最基本的特征,也是判别市场是否存在泡沫的重要依据。

1637 年荷兰的郁金香泡沫,郁金香球茎 1 年总涨幅高达 5 900%,单位价格超过稀有宝石;1719 年南海泡沫,公司股票从 129 英镑狂飙至 1 000 英镑以上,半年涨幅达到 700%;1990 年日本房地产泡沫,仅东京的地价就相当于美国全国的土地价格。很显然,这些资产的价格已经疯狂地背离了本身的价值,但是投资者丝毫不关心,他们的标准只有一个:价格能涨就行。他们买入金融资产的原因仅仅是预期有另一个人会出价更高,会有更傻的人以更高的价格从他们手中接过“烫山芋”,形成了典型的“博傻”模型(theory of greater fool),整个过程与价值没有任何关系。因此,市场价格巨幅上升时,它在投资者之间创造了大量的成功神话,这些神话吸引潜在的投资者,他们天真地认为同样的成功会降临在他们头上。他们往往没有投资经验,看到市场利好便纷纷涌入,想分得一杯羹。在这种情形下,即使金融资产的价格已经被炒到不合常理,投资者仍然不会收手,杠杆使得资产价

格与价值的偏离程度进一步扩大，结果就是泡沫化。

2）交易量异常持续放大

泡沫的膨胀期，交易量会异常持续放大。一方面，金融资产价格的持续上涨会吸引大量投资者资金投入，并在上涨的过程中不断引发更多资金的追涨；另一方面，新资金的注入令泡沫进一步膨胀，一步步形成“正反馈”。

专栏 11.1 我国权证市场的泡沫

2005 年 8 月，中国证监会(CSRC)首次推出权证，迈出发展金融衍生品市场的重要一步。2008 年 5 月，18 只认沽权证和 29 只认购权证向公众推出，其中，32 只在上交所交易，15 只在深交所交易。本次推出的权证产品有以下几个特点。

(1) T+0 交易制度。

(2) 禁止卖空机制。

(3) 免交印花税和注册费，交易费用更低。

(4) 权证涨跌幅=标的股票涨跌幅×125%×行权比例，限制很小。

权证一经推出，成交价便大幅增长，日均成交量达到 12.6 亿元，平均年收益波动率为248%。以产品“五粮 YGP”为例，“击鼓传花”式交易后价格居高不下。据测算，权证成交价格远远高于 BS 公式定价，甚至在某一时期超过了行权价，而看跌期权的内在价值是不可能大于行权价的。交易期间，“五粮 YGP1”的日换手率平均为 140%，最后一个交易日达到了 1 800%。这意味着，平均每份权证被买卖了 18 次，如图 11.2 所示。

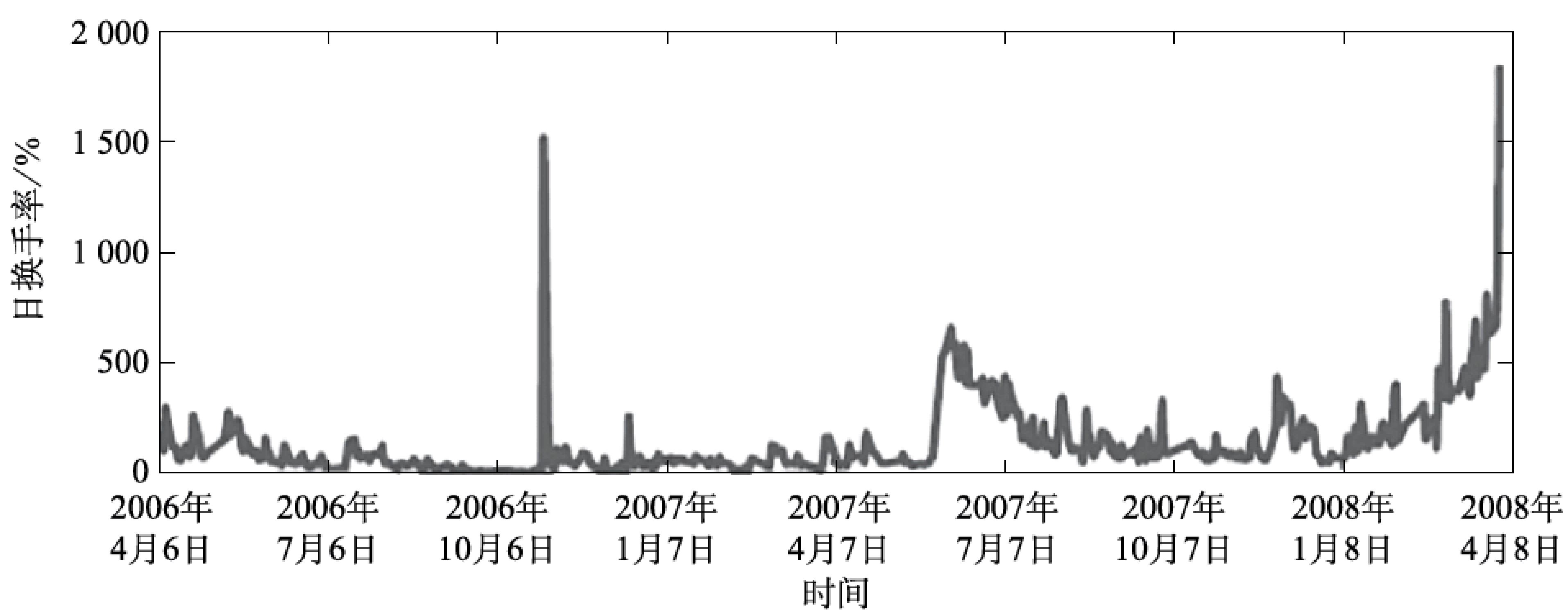


图 11.2 “五粮 YGP1”日换手率

资料来源：XIONG W，YU J. *The Chinese warrants bubble*[J]. *American Economic Review*，2011，101(6)：2723-2753.

3）大量低认知水平投资者涌入

有这么一个故事：19 世纪 20 年代，美国股市一片繁荣，整个国家都处于牛市氛围。有一次，美国证券交易委员会首任主席约瑟夫·肯尼迪在纽约中心火车站一边读报，一边让一个擦鞋童给他擦皮鞋，这个擦鞋童告诉约瑟夫，他也买进了一些股票，有一个要发财的美国梦想。擦鞋童的话，让约瑟夫意识到，19 世纪 20 年代的股市牛市行情结束了。他的理由是：“如果连擦鞋童都知道进入股市买股票，那还剩下谁没进入股市呢？既然大家都已经跳进了股市，那市场就只能是下跌，没有其他路可以走了。”于是，约瑟夫·肯尼迪把

他的资金从股市中撤出，从而躲过了 1929 年股灾。这个故事表明，当无论哪行哪业的人都不顾一切地冲向一个市场的时候，就说明市场的泡沫已经被吹得太大了，市场基本上已经到了顶部。

美国经济学家 DeLong 等曾提出一个著名的噪声交易者理论，这个理论假设市场上除了根据基本面价格进行交易的理性交易者以外，还存在大量非理性的噪声交易者。和理性交易者不同，噪声交易者一般不太关注资产价格的基本面信息，而是提交一些想当然的报价和交易数量，或者采用简单的追涨杀跌式的跟风交易策略。大量噪声交易者涌入市场通常会损害市场的稳定性。包括美国亚利桑那大学查尔斯·纽赛尔和德国柏林工业大学的博施-罗沙在内的许多学者都在实验室模拟情境下发现了交易者的认知水平的确会影响他们的交易策略和实验市场的价格稳定性。交易者认知水平越低，市场越容易产生泡沫。也就是说，在泡沫经济中，往往有低认知水平的非理性交易者的推动。

3. 金融市场泡沫的测度①

学术界对于股市泡沫的研究经历了由理性泡沫到非理性泡沫的过程，从而形成两个学派。理性泡沫是指在理性预期的框架内，市场价格相对于资产未来各期收益现金流的折现值(理论价格，也称基础价值或合理价值)的偏离。非理性泡沫模型侧重于采用人的非理性因素(群体心理、时尚、狂热)等来解释理性泡沫模型难以解释的金融市场极端波动的现象。无论从什么角度来解释泡沫，首先都需要对泡沫进行度量。主流的对泡沫的度量方法可以大致分为三类：基本面估计法、统计量检验法和价格序列分解法。

1) 基本面估计法

在理性泡沫中，泡沫通常被定义为理性预期条件下市场实际价格与内在价值(理论价格)之间的差别。所以，只要能够计算出资产的内在价值，那么，股市中的价格泡沫基本上就是可度量的。然而，要想估计一项资产的内在价值，就意味着必须对该项资产的预期收益和未来利率水平作出假定。如果这些前提假设出现问题，那么对该项资产内在价值的计算就会存在偏差，如用股利现值定价模型(戈登公式)来判断是否存在泡沫就是非常困难的一件事。因此，有不少学者通过对市盈率、周超额回报率、工业生产指数、居民消费价格指数、狭义货币供应量等与股市泡沫相关的指标的计算，或通过相关指标的数理统计特性，表征不同时点股市泡沫大小的比较关系，以分析股票市场价格中的内在价值部分。

潘国陵(2000)以过去的每股税后利润预测未来的每股税后利润，又以预测的每股税后利润替代每股红利，并运用股息折现模型计算上海和深圳股市 1997—1999 年的内在价值，进而测算相应年份的股市泡沫。吴世农(2002)等在假设我国股票市场派息率为零的情况下，运用 CAPM 模型来确定股票市场的内在价值，然后计算股市泡沫的大小。徐爱农(2007)则用剩余收益模型对中国 A 股市场 1993—2006 年年初市场内在价值进行计算，并对相应年度年初市场泡沫进行测度。

基于基本面的度量方法需要得到能够反映经济基本面因素的经济变量，但是与股市

① 主要参考周爱民，孟庆斌．大鱼如何吃小鱼：股市价格泡沫的度量与理性扩容速度的行为金融学分析[M]．厦门：厦门大学出版社，2009.

价格相配合的日数据和周数据很难得到，这是该方法的局限性。

2）统计量检验法

席勒（1981）提出了超常易变性方差检验方法，他在假设 H_0｛不存在股价泡沫｝之下，试图通过比较理论股价和实际股价的方差之比来进行关于价格泡沫的 F 检验。他认为 $\text{Var}(P_t)\leqslant\text{Var}(P_t^*)$ 是应该成立的，因为：$P_t=E(P_t^*\mid I_t)$，所以在掌握一定信息情况下变量的不确定性一定要小于不掌握任何信息情况下该变量本身的不确定性，所谓的超常易变性统计量：$\text{Var}(P_t)/\text{Var}(P_t^*)$，就是根据这一假设给出的。但是理论估计是很难得到的，因此他转而借助估计值来替代：

$$P_t{}'=\begin{cases} P_T & t=T\text{ 时}, \\ \sum_{v=1}^{T-t}\dfrac{d_{t+v}}{(1+r)^v}+\dfrac{P_T}{(1+r)^{T-t}} & t=0,1,2,\cdots,T-1\text{ 时} \end{cases}$$

该方法曾运用于以 S&P500 指数为样本的检验，检验出许多股市泡沫。结果如图 11.3 所示，其中实线为美国标普 500 真实值，虚线为预测值。

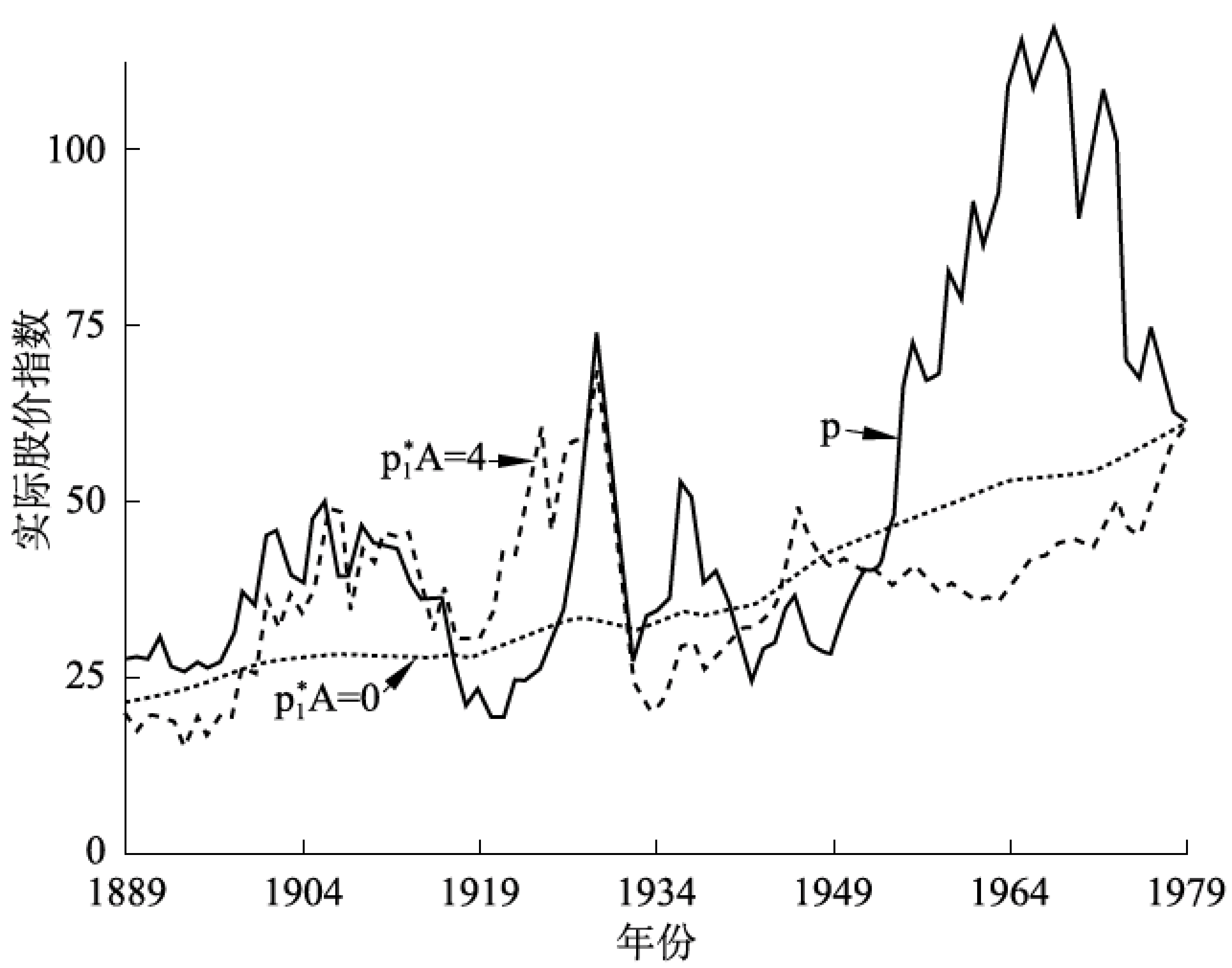


图 11.3　1889—1979 年美国标普 500 指数真实值与预测值

但是后续 Kleidon（1986）等对该方法提出了以下几点质疑。

（1）用可计算的估计值来替换不可计算的理论价值会使得该方法的有效性遭到破坏。

（2）当统计量拒绝原假设的时候，并不一定是因为股价泡沫的存在，即该检验方法接受假设的结论。

（3）统计量检验的方法经常会给出大量的伪信号，检验过程很复杂。

除了超常易变性方差检验方法以外，Engle（1987）也曾提出协整检验方法，试图通过对股价过程和分红过程间的协整性检验来确定价格泡沫的程度。Diba（1988）发现如果存在理性价格泡沫，那么股票价格的一阶差分应该是非理性的。Craine（1993）提出，如果市盈率存在单位根，那么就意味着存在"非理性繁荣"。

3）价格序列分解法

价格序列分解法致力于从价格序列中分离出理性价格来，大致可以分为两分法和三

分法。前者可以将价格序列分离成理性价格与剩余项，后者还能在剩余项中分离出理性泡沫项。下列式子是两分法的形式之一：

$$P_t = \lim_{T\to+\infty} p_t = \lim_{T\to+\infty} \frac{p_T + t}{(1+\theta)^T} + \lim_{T\to+\infty} \sum_{j=1}^{T} \frac{d_{t+j}}{(1+\theta)^T} = b_t + p_t^*$$

其中，p_t^* 为理性价格；b_t 则可以被定义为价格泡沫。这样定义的泡沫有以下两个特点。

(1) 任何一天的价格泡沫都能被其后一天的价值所证实。

(2) 价格泡沫不可能为负，否则可能会出现价格为负的情况。

4. 金融市场泡沫破灭的危害

"1929 年 10 月 24 日，黑色星期四，美国股票市场崩溃。成千上万的美国人，从一贫如洗的寡妇到腰缠万贯的大亨，都失去了他们终生的积蓄。到这天结束时，已经有 11 个金融家自杀。从这个角度看，来往穿行于华尔街的人们像一群热锅上的蚂蚁……胡佛总统打电话给参议员波默林，任命他主持复兴金融公司。在接到电话时，波默林的口袋里只有九角八分钱，而且在他前往宣誓就职的路上，就有 10 个叫花子跟他要钱。"①《光荣与梦想》里的这段话形象地描述了美国金融市场泡沫破灭后的大萧条景象，10 万家企业破产，全国 1/4 的人口失去工作，无数的富豪、投机者沦为乞丐。麦当劳创始人雷·克洛克被迫做了 17 年的纸杯推销员来还债，《股票大作手回忆录》里的主角杰西·利弗莫尔吞枪自杀。

泡沫的破灭一方面使得参与投资泡沫的个人财产受到极大的损失，消费需求长期萎靡，大量资金投入估值过高、投资无效的市场里，降低了资金的使用效率。更严重的是，它往往会导致金融危机的爆发，企业因为过于膨胀的负债而纷纷倒闭，社会消费与投资需求下降，又引发通货紧缩和经济衰退，给一国经济带来巨大的危害。严重的房地产危机、股市危机、银行业危机还会通过进出口贸易、跨国投资等途径席卷全球，最终会演化为社会危机、政治危机甚至引发军事危机。

11.2 金融市场泡沫的标准金融学解释

通过前面的学习，读者已经初步地了解了金融市场泡沫的含义。但是，市场中究竟有没有泡沫存在，以及为什么有泡沫存在仍然是困扰读者的一个问题。因此，本节主要从标准金融学角度来解释金融泡沫。

1. 完全理性理论

根据有效市场假设，如果交易者都是完全理性的，则他们所掌握的信息就会充分体现在对证券基本价值的估计中，证券的实际价格将会等于其基础价值。这是因为，理性交易者基于效用最大化的投资行为所决定的证券价格是正确的，恰好等于证券的基本价值，一旦市场价格偏离了证券的基本价值，理性行为主体就会通过套利策略获利，所以在有效市场的框架内是不存在泡沫的。

① 曼彻斯特. 光荣与梦想:1932—1972 年美国社会实录:上[M]. 海口:海南出版社,2004.

2. 理性泡沫理论

理性泡沫理论以市场有效性为前提，认为在理性行为和理性预期的假定下，金融资产的实际价格除反映其市场基础价值之外，还包含理性泡沫的成分，这种泡沫并不是错误定价的结果，而是理性预期和理性行为允许的资产价格对基础价值的偏离，即反映了未来。

理性泡沫理论遵循了主流经济学的研究方法，通过建立模型来证明在标准金融理论的框架内，即使个体的行为都是理性的，总体结果也可能导致证券价格与基本价值的背离，即人的理性行为也会产生泡沫。

在理性泡沫框架下，如果市场参与者是有限的，则市场不可能形成泡沫，这主要是因为理性投资者只有在确信将来能通过卖出证券获取收益的情况下才会买入证券，当投资者数目有限时，投资者难以确定在价格上升之后是否能够找到买方，因而排除了泡沫存在的可能性，如表 11.1 所示。

表 11.1 一般均衡框架内的理性泡沫存在性①

参与人 \ 交易时间	无限交易时间	有限交易时间
有限市场参与人	不可能存在泡沫	不可能存在泡沫
无限市场参与人	可能存在泡沫	可能存在泡沫

理性泡沫理论的主要缺陷是对泡沫过程缺乏微观机制的描述，泡沫的生成和破裂基本上被看作外生的，没有很好地解释泡沫生成的条件、机制和导致泡沫破裂的事件。理性泡沫理论虽然是难以令人信服的，但至少意味着主流学者意识到了市场价格会持续偏离基础价值，有效市场假说的结论是不恰当的。

近年来的众多实验经济学研究发现人类的非理性交易行为容易使资产价格产生泡沫。经济学家开始放弃有效市场的假定，运用行为金融学以及博弈论等研究股票泡沫，形成了非理性泡沫理论。和投机性泡沫理论不同的是，该理论更多的是从交易者个体心理和外部因素相结合这一角度研究什么因素导致了股票泡沫。非理性泡沫理论将市场噪声和投资者差异纳入考虑范围，发现投资者噪声交易、羊群行为、有限关注等非理性行为一旦形成一致性偏差，会导致资产价格对价值系统性的偏离，推动泡沫生成。

专栏 11.2 投资者认知能力与金融泡沫——来自实验室的证据

纽赛尔在实验室中用美国心理学家弗雷德里克提出的认知反射测试来衡量被试的认知能力。该测试由三个智力测验题构成，如："一个球拍和一个球共计 1.1 元，球拍比球贵 1 元，请问球的价格是多少钱？"

如果回答者未经思考直接回答，通常会说 0.1 元，但实际答案是 0.05 元。实验组织者根据回答者的得分把他们分为高认知能力者和低认知能力者，回答正确的问题越多，表示

① TIROLE J. *Asset bubbles and overlapping generations*[J]. Econometrica, 1985, 53(6): 1499-1528.
WEIL P. *Confidence and the real value of money in an overlapping generations economy*[J]. *Quarterly journal of economics*, 1987, 102(1): 1-22.

认知能力越高。然后,实验组织者会把回答者分配到各个实验市场中去。市场一般分为三类:全部由高认知能力者组成的市场、全部由低认知能力者组成的市场和两者都有的混合型市场。在每个市场里,交易者都知道自己市场中高低认知能力的参与者的组成比例。

实验的结果显示主要由高认知能力交易者组成的市场一般没有泡沫,主要由低认知能力交易者组成的市场泡沫较大,而由一半高认知能力交易者、一半低认知能力交易者组成的市场泡沫最大。这个结果表明:认知能力确实是金融泡沫的重要决定因素。如果市场全部由高认知能力者组成,他们的交易策略一般会比较理性,不容易发生泡沫;而在由低认知能力者组成的市场里,交易者可能由于认知能力的局限更容易采取投机型的交易策略,从而造成泡沫。在混合型市场中,交易者可能面临更大的策略不确定性,且高认知能力交易者可能通过误导低认知能力交易者赚钱,所以价格更不稳定了。

为了验证在实际市场里,交易者预期对于资产价格也有重大影响,宗计川等在东北财经大学做了一个实验,进一步检验认知能力对于个体预期策略和模式的影响。

他们发现:如果通过认知反射测试把交易者分为高认知能力者和低认知能力者,高认知能力者在市场中更容易采用有利于市场价格稳定的适应型预期,而低认知能力者在市场中更容易采用催生泡沫的趋势跟踪型预期,也就是人们常说的"跟风"。CRT 得分低的交易者不但更可能采取跟风的预期策略,而且当他采取这一策略的时候,他跟风的"力度"也更大。

上述研究都表明:认知能力对于交易者策略和市场稳定性都有巨大影响,低认知能力的投资者往往会作出非理性的投资决策,催生市场泡沫。

资料来源:宗计川,付嘉,包特. 交易者认知能力与金融资产价格泡沫:一个实验研究[J]. 世界经济,2017,40(6):167-192.

11.3 金融市场泡沫的成因

1. 宽松的货币政策

根据经济史学家的总结,人类历史上任何一个泡沫的背后,无一例外都伴随泛滥的流动性。荷兰、英国、美国、日本这些一度的全球经济霸主,都无一例外地在其经济发展上升期,经历过不同形式的泡沫和崩盘。这主要是因为在经济发展上升期,货币政策制定者往往会采取相对宽松的货币政策,同时周边地区乃至全球的资本都会流入当地,进一步导致了流动性大泛滥,从而引发资产价格泡沫。越来越低的实际利率会导致资金大量流入金融与地产领域,吹大了资产价格泡沫。

诺贝尔经济学奖得主、经济学家史密斯的实验研究表明,如果资产总价值是一个定数,那么实验参与者拥有的钱越多,也就是整个实验市场里面的流动性越泛滥,实验市场里泡沫形成的规模就越大,持续的时间也越长。正如朱宁(2014)所指出:"因为流动性如此之大,钱多到了任何安全资产的回报都低得让投资者不愿意接受的地步(比如美国2007—2009 年房地产危机之前的国债利率)。投资者为了追逐让他们更满意的高收益,就不得不寻求其他高风险的投资机会。在对收益的追逐过程中,人的贪婪就会一次次地压

倒人的恐惧,驱使着投资者卷入一次又一次的市场波动之中。”[①]

这在股票市场尤为明显,因为股市中高流动性的优势会使其更容易吸引巨额资金。历史上诸多股市泡沫的膨胀过程都离不开充裕流动性的支持。

1997 年亚洲金融危机爆发后,大量国际资本回流美国,而美联储 1998 年 9 月至 11 月连续 3 次降息,进一步拉低了资金成本,促成了美国股市互联网泡沫迅速膨胀。

2008 年次贷危机爆发前,美联储将联邦基金目标利率从 2000 年 5 月的 6.5%一路下调至 2003 年 6 月的 1.0%,直至 2004 年 6 月才启动加息。长期的低利率政策大幅拉低了融资成本,这是美国房地产市场爆发性增长的重要原因,也为次贷危机埋下了伏笔。

我国 2007 年 A 股泡沫的产生也与国内流动性密切相关。2005 年汇改启动之后,人民币稳步升值,一方面对外贸易持续高额顺差;另一方面国际资本通过各种渠道涌入我国,导致金融机构外汇占款余额飙升。2005 年 6 月,外汇占款余额仅为 6.26 万亿元,到 2007 年 9 月底,外汇占款余额已达 12.57 亿元。为了对冲过多的流动性,央行一方面大幅提高存款准备金率;另一方面大量增发央行票据,2005 年 6 月底,央行票据余额仅 1.65 万亿元,到 2007 年 9 月底已达 3.90 万亿元。

2015 年我国股市泡沫的重要原因之一便是 2014 年以来持续的宽松货币政策。2014 年至 2015 年 6 月股市暴跌前,一年期存款基准利率由 3%下调至 2.5%,存款准备金率由 20%下调至 18.5%。从时间点来看,股市暴涨开始的时点与 2014 年 11 月首次降息开始货币政策宽松周期的时点是一致的。流动性充足与实体经济投资机会缺乏,导致了“资产荒”的出现。由于坏账率飙升、房地产投资收益下滑等原因,大量银行资金不愿投向信贷,而是通过场外配资、融资融券等方式进入股市,成为股市加杠杆的主要资金来源,是导致股市泡沫的重要原因。

2. 新技术引入

泡沫往往与新技术有关,新技术蕴含的较高的不确定性可能导致不当的监管和过高的估值水平。一方面,新技术,如某些金融创新工具本身可能会存在缺陷;另一方面,短时间内,人们对新技术的认识是有缺陷的。正如朱宁指出:“泡沫的产生,首先需要有新事物的出现。泡沫的诞生往往和一种新事物、新产品、新技术,或者新的经济地区相联。由于新的技术、产品、市场的出现,人们就很难用传统的方法或者维度来评价新鲜事物。而对传统体系的否定,和对历史规律和经济学原理的挑战,就会给‘新思维’‘新模式’提供成长的空间。”

正是因为人们难以用传统的理念来解释,泡沫才会在一段时间内保持其神秘性,而这无疑会激发更高的投机热情,引发投资者希望获得更高的收益。又由于投资者不愿意了解或承受高收益背后的高风险,或者由于这种风险本来就无法在崩盘之前估量,投资者这样一种天真而美好的“一本万利”甚至“无本万利”的愿望,往往就成为泡沫形成最好的铺路石。

① 朱宁. 五大因素催生和吹大经济泡沫. 东方财富网.

专栏 11.3 金融产品创新与美国次贷危机

2007 年美国次级贷款泡沫产生的一个很大的动因便是资产证券化等创新衍生产品的出现。美国金融机构向低收入人群发放了大量的住房抵押贷款后，为了获取更好的流动性，分散和转移风险，将购买到的贷款打包成证券，即住房抵押贷款证券(MBS)，兜售给投行等金融机构和其他投资者。投行购买到 MBS 后，对基础的现金流进行重组，设计出风险和收益不同档次的担保债券凭证(CDO)，卖给各地的投资者。为了降低风险，又推出了能对冲低质量档次 CDO 风险的信用违约互换(CDS)产品。

这一系列衍生产品的发行，使得次级抵押贷款这一基础资产上面垒筑起越来越高的衍生品。这些金融衍生品经过层层的打包重组之后，杠杆倍数被放大到了极限。投机者大量进入衍生工具交易市场，形成巨大的次贷衍生品泡沫。随着房地产市场的降温和加息周期的到来，基准利率的上升使得抵押贷款利率随之上升，债务人无法按时偿还债务，次级贷款违约率不断增加，MBS 和 CDO 的回报也失去了保障，对冲基金亏损日益严重，再加上先知先觉的投资者开始"用脚投票"，大量次贷衍生品的价格一泻千里，直接导致这些对冲基金的资金链断裂，最终引发了次贷危机。对冲基金的溃败很大部分是因为它们认为自己已经很好地对冲了风险，然而对冲工具的设计者并不能理解复杂衍生品的风险所在，其设计的对冲产品收益和风险严重不对称，一旦对冲产品因为市场波动不足以覆盖风险，则会进一步导致被对冲机制掩盖的金融市场系统性风险集中暴露。所以，复杂的金融衍生产品有着价格信号失灵、信息不对称、监管缺位、信用对冲等一系列天生的缺陷，它的泡沫破裂只是时间问题。

资产证券化本身的复杂性给参与主体带来了认识上的困难。美国的次级贷款进行过多次的证券化，并运用了复杂的分档、错位、叠加、合成、放大、对冲等技巧，使资产证券化的过程变得更加扑朔迷离。这种证券化的复杂性至少造成两个后果：一方面，它使得证券化的参与方以及监管机构对房贷信用风险的影响估计和监控变得十分困难；另一方面，产品的复杂化使得投资者难以凭自己的认知去匹配风险和收益，对金融资产进行过高估价，或者对风险认识不足，以致对风险视若无睹，使其不断积累。

3. 代理投资[①]

现实中大量投资行为包含委托代理关系，真正的决策者往往是投资的代理人，与全部使用自有资金投资不同，代理投资带来的一个问题是，作为代理人的投资决策者可以享受到资产价格(收益)上升带来的全部好处(upside return)，但是对资产价格(收益)下跌的风险(downside risk)只承担有限责任，投资代理人可以通过申请破产保护等方式将超过一定限度的损失转嫁给投资委托人——资金的贷出方，这就是代理投资内生的风险转嫁(risk shifting)问题。以房地产为例，当投资成功时，借款人归还贷款本息，房地产的价值超过银行本息的部分就成为借款人的投资收益，当投资失败时，借款人只承担有限责任，贷款人必须承担其余的损失。当市场上有相当一部分投资决策者是投资代理人时，

① 主要参考陈国进，吴锋．代理投资、金融危机与金融制度结构[J]．金融研究，2002(8)：61-66.

他们内生的风险转嫁激励就会使风险资产的均衡价格超过基本价值，从而催生资产的价格泡沫。

4. 媒体偏差[①]

公共新闻媒体是一种重要的信息传递方式，媒体能积极地影响公众的注意力和思考方式，从而形成股市事件发生时的环境，众多投资者对经济信息的感知往往通过媒体报道所描述的拟态环境来获取，并依据环境的变化来作出判断，调节自身行动，进而作出投资决策。

媒体报道对资产价格的影响主要通过两个渠道实现：一是进行信息披露，媒体的报道会向市场提供反映公司价值的最新信息从而影响资产价格。但是媒体不可避免地会出于自己的利益或者政府政策倾向的压力而对释放的信息进行甄别和筛选，导致媒体偏见的形成。二是汇集和扩散信息的中介作用，在投资者"有限关注"的前提下，媒体报道可以提高投资者对被报道公司的关注度，缓解投资者因关注度不足导致的分散化投资不足问题，进而降低资产要求的风险补偿。如 1998 年 5 月 3 日，《纽约时报》报道 EntreMed 公司研制出一种非常具有市场潜力的新抗癌药物，导致该公司的股价从 12 美元迅速上涨到 52 美元，并且在接下来的 3 周内一直稳定在 30 美元以上。但是这项研究成果早在 5 周前在《自然》杂志上就已经公开发表了。

张圣平等(2014)利用百度新闻搜索得到的媒体报道数据，发现财经媒体报道对中国股票盈余惯性有着显著的影响。盈余公告之后的媒体报道虽然客观上已不包含任何新信息，但仍可引导投资者对所报道股票的注意力，影响股票的价格和交易量。首先，媒体报道加快了利好信息进入价格的速度，减小其盈余惯性；其次，媒体报道增强了多空双方的信念，公告后异常交易量上升。与此同时，媒体报道有选择性，常顺应市场的反应对利好信息"锦上添花"，没有过多地"落井下石"。媒体的正向选择性报道会促使投资者对市场充满信心，参与投资，推动泡沫的膨胀。

另外，为了吸引更多投资者的关注，媒体会对信息进行加工，过滤一些有用信息或者强化某些信息，使得媒体本身不再是一个独立传播信息的中介，甚至在某些时候会成为推动市场趋势的力量。尤其是在现代社会，互联网、微博与微信等新媒体方式在人群之中迅速传播，资本市场的魔力和泡沫形成的赚钱效应被无数倍地放大了。在媒体的感召之下，很多投资者被卷入一场又一场的疯狂游戏之中，媒体的乐观情绪甚至成为投资者坚信资产价格会继续上行的背书。

如 2014 年年中到 2015 年年中我国股市大幅异常增长的时期，除了来自经济转型预期的支撑，官方舆论关于股市的表述也往往被社会舆论解释为替股市上扬背书。其中比较有代表性的包括：2014 年 8 月，A 股在开始上涨不久后陷入横盘震荡，8 月 18 日，《人民日报》刊文《股市保持强势 望演绎空中加油行情》；2015 年 1 月 5 日，新华网刊文《华尔街专家：中国 A 股今年将继续走牛》；2015 年 4 月 21 日，人民网刊文《4 000 点才是 A 股牛市的开端》等(表 11.2)。这些无疑都使得投资者更确信价格会继续上行。

---

① 主要参考管涛，邓海清，韩会师．泡沫的终结——透视中国股市异动[M]．北京：中信出版社，2016.

表 11.2　2014—2015 年我国主流媒体对股市的相关报道

类别	时间	内容
牛市导向	2014/07/25	《人民日报》:外资机构齐声唱多 A 股市场
	2015/01/05	新华网:华尔街专家称中国 A 股 2015 年将继续走牛
	2015/04/08	新华网:A 股中长期仍有上涨空间和动力
	2015/04/10	人民网:A 股 4 000 点上升的基本动力不变
	2015/04/21	人民网:4 000 点才是 A 股牛市的开端
	2015/05/08	《人民日报》:管理层治理旨在慢牛,长牛势头未变
	2015/05/28	新华网:6 月股市将走慢牛行情
风险提示	2014/12/19	《人民日报》:多重因素驱动 A 股牛市,短期仍存在调整风险
	2015/04/09	新华网:潮来潮去终退去,且行且珍惜
	2015/05/04	《人民日报》:牛市也别忘风险

11.4 个体心理与金融市场泡沫

1. 有限关注

有限关注最早由卡尼曼(1973)提出,他认为个体的认知能力是有限的,需要在不同的事物上分配认知资源,分配在一个事物上的注意力必然减少个体对于另一事物的注意力,并称这种现象为有限注意。

传统的有效市场理论认为市场参与者具有完美理性和无限信息处理能力,能够及时充分地将所有公开信息反映在股价中。然而,真实的市场由许多有限注意的投资者组成,他们容易受到媒体宣传和市场情绪的影响,特别是在互联网时代的今天,投资者更是面临空前的"信息爆炸",大量新闻、消息、评论甚至谣言充斥网络渠道,使投资者不得不选择性地关注某些信息,而忽略对其他信息的处理。"当今经济社会的信息富裕问题引起了新的稀缺,因为海量信息消耗了人们大量的注意力,造成了注意力的贫穷。随着互联网技术的进步和媒体作用的加强,投资者在进行资产组合决策时所遇到的问题往往不是信息稀缺,而是信息过多和信息处理能力不足。"[①]

Baber 等(2008)认为投资者在买入股票时面临成千上万种股票选择。由于人的认知能力有限,只有那些吸引投资者关注的股票才会进入投资者选股的考虑集。也就是投资者尤其是个人投资者更倾向于购买那些引起他们关注的股票。而当面临卖出决策时,受卖空约束的个人投资者不会面临严重的注意力约束问题,因为他们通常只持有少数几只相对熟悉的股票。这种注意力约束对买卖决策的非对称影响势必导致投资者对那些被关注股票产生净买入行为。这就会引发这些受关注的股票因为过度的需求而产生泡沫,其

① 彭叠峰,饶育蕾,王建新. 有限注意、投资者行为与资产定价——一个研究评述[J]. 中南大学学报(社会科学版),2012,18(3):116-122.

他金融资产泡沫也是一样的道理。

2017 年 4 月 1 日，我国政府印发通知，决定设立河北雄安新区，在随后 1 周内引发资本市场的强烈关注，部分雄安新区概念股快速上涨甚至在短短 1 周内已出现过度投机炒作苗头。如若不是上交所迅速地干预监管，责令相关上市公司作出澄清，发布异常波动公告，集体对市场传闻予以说明澄清，这一概念股的泡沫会持续得更久、膨胀得更大。然而，早在此通知发布半年前，包括央视等媒体已经报道过政府领导赴河北省安新县进行实地考察、主持召开河北雄安新区规划建设工作座谈会等消息。只是这些信息没有被炒热，引起市场所有投资者的注意力而已。

另外，当股价由于某些原因上涨时，投资者由于有限注意而关注上涨的股票，导致对这些股票的需求增加，促使价格进一步地上涨，进而又引起其他投资者的注意，导致股价严重超过基本价值，而如果这样的有限注意成为系统性的群体行为，就可能导致股市泡沫的产生。

在我国股票市场，投资者注意力所诱发的股票市场波动和炒作的其他现象也比比皆是。例如，2005 年中国证券市场的第一只权证——宝钢权证引起了投资者广泛的关注和热情，不到 3 个月，上市的 6 只权证激发了超过 100 亿元的单日市场成交额，远超过同期的沪深两市股票的成交总额。Liu 等(2012)发现权证的引入带动了相应标的股票的高波动率和成交额，这种投机的溢出不是基于信息，而是广大投资者热切关注所驱动的市场炒作。此外，中国股市针对一些重大的社会事件，如 2003 年 SARS(重症急性呼吸综合征)爆发和 2008 年的四川地震等，会产生强烈的行业联动乃至市场震荡效应，这种过度的联动是无法完全用信息本身来解释的。

2. 过度乐观

在乐观情绪中，市场存在“自我实现的预言效应”，即对未来市场预期价格上升，现实价格也会随之上升；反之，若对未来市场预期价格下降，现实价格也会随之下降。其显著的特点是预期的同质性，即经济主体对资产价格的走向具有共同的预期。正反馈交易理论认为最初的价格上涨会导致更高价格的上扬，或最初价格的下跌会导致更低价格的下降，即人们通常所说的追涨杀跌。由于投资者需求的增加，最初价格上涨的结果又反馈到了更高的价格中，信息对资产价格的初始影响被放大。

由于投资者过于乐观的情绪，使得在金融资产价格畸高的时候，投资者仍然坚信价格还会上行，总会有人以更高的出价来接手自己买入的金融资产。这也是为什么在金融资产跳崖式下跌前市场总是一如既往的狂热。正是因为大家都觉得没有泡沫，泡沫才有可能成为泡沫。如果大家都觉得经济处于泡沫而缩减自己的投资和风险敞口，泡沫也就不会成为泡沫了。

类似地，在美国、日本、我国香港地区的房地产泡沫期间，一阵阵“这一次不一样了”的说法，让多少个人、家庭、机构忘记了房价与房租比例、房价与收入比例这种简单的投资基本面的评价方法，而盲目地相信房价迟早会涨到天上。金融市场泡沫的一个重要特征就是不关注基本面，而那些关注市盈率、市净率等传统估值方法和坚持这种投资理念的像股神巴菲特这样的投资者，则在互联网泡沫时期被嘲笑为“食古不化”而“很快就会被历史淘

汰”。但现实是，巴菲特还没去世，很多嘲笑他的互联网公司早就破产倒闭了。

3. 羊群行为

关于个体暗示和群体压力的心理学研究表明：个体的决策受其他人意见所影响，社会压力能诱导个人作出自己认为并不正确的决策。在存在不确定性的环境中，人们通常表现出一种群体行为，即在某段时期内同向买卖相同或者相似的金融资产的一种模仿行为。泡沫中的羊群行为往往是自发形成的、表现出非理性的特征，从而导致了意见和行动的传染。

股票等金融资产的价格也不外乎是社会活动的结果。席勒(1984)发现因为没有一个可以接受的理论来理解股票的价值，也没有可靠的方法来预测改变投资策略后的效果。投资者对股票的价格缺乏一个清晰、客观的认识，从而致使他们完全依赖社会活动过程作出投资决策。Topol(1991)发现个体投资者买卖股票通常按照他最熟悉的投资者的平均价格来报价，由此可以推测股票价格运动是通过社会来传递的。

声誉模型也非常恰当地解释了泡沫时投资者的心理。与一个另类而可能成功的策略相比，人们更愿意表现出羊群行为，成为群体失败的一员。在美国互联网泡沫破灭之前，坚持巴菲特“价值投资”理念的投资者长期被嘲笑是食古不化的落后人群，人们不愿意成为被嘲笑的另类，哪怕自己可能才是会成功的少数派。尤其是基金经理们，他们的业绩随时会被投资者用来与其他基金经理相比较。如果他们逆流而动却失败了，就会被认为能力不够，受到委托者的责备。但如果他们的行为与大多数人的一致，那么即使市场下跌，委托人也会考虑到其他人同样失败了而不去苛责他们，使其避免承担决策错误的责任。因此，即便很多精明的专业投资人，也总是试图“与泡沫一起膨胀”，而不是努力避免泡沫，在价格上涨过程中，他们通常认为随大流比采取与众不同的方法更安全，循规蹈矩比特立独行犯错误的可能性更小。在 1987 年 10 月崩盘前的牛市，专业资产管理者都觉得当时的价格水平太高了——股市更可能下跌而不是上升，然而没有多少资产管理者愿意卖掉他们的股票。股票继续上涨，他们害怕由于错失良机而被看作少有的傻瓜。相反，即使市场下跌，他们也会由于人数众多而感到安慰。如果每个人都遭受同样的厄运，大家的命运就是一样的。①

从众心理会导致投资者不计风险地持有过度高估的资产。即使金融资产价格已经飙升到一个疯狂的高位，从众的投资者也仍然不会选择减持，继续维持着市场旺盛的需求，一步步助推泡沫不断膨胀，直到泡沫破灭。

专栏 11.4 代理投资下资产价格泡沫的形成

假设有两种资产——安全资产和风险资产，安全资产的供给是可变的，安全资产的期初价格为 1，期末价值为 1.5；风险资产的供给固定，风险资产期初的价格为 P，期末价值为 6 的概率为 0.25，期末价值为 1 的概率为 0.75，风险资产期末预期价值为 2.25；所有的参与者都是风险中性者。

---

① FROOT K A，SCHARFSTEIN D S，STEIN J C. *Herd on the street：informational inefficiencies in a market with short-term speculation*[J]. *Journal of finance*，1992，47(4)：1461-1484.

如果每个投资者都拥有 1 个单位的资金，可以将它投资于安全资产或风险资产，由于投资者为风险中性者，当资产市场处于均衡状态时，投资于这两种资产的边际收益相等，即 2.25/PF＝1.5/1，整理后可得 PF＝2.25/1.5＝1.5，公式中 PF 为用自我资金投资时风险资产的价值，换言之，PF 为风险资产的基本价值，风险资产超过基本价值的部分就是泡沫。下面考察代理投资下资产价格的泡沫是如何形成的。

假设投资者借入 1 个单位的资金，借款利率为 33%，贷款人与借款人之间就形成了投资的委托代理关系，如果贷款人不能观察到借款人(投资者)究竟是投资于安全资产还是风险资产，那么 1.5 是否还是风险资产的均衡价格呢？

投资者将借入的 1 个单位的资金投资于安全资产的收益为 1.5－1.33＝0.17。如果风险资产的价格为 1.5，投资者在期初可以购买 1/1.5 个单位的风险资产，当期末风险资产的价值为 6 时，归还贷款本息后剩下部分为投资者的收益；当期末风险资产的价值为 1 时，风险资产价值全部归贷款人所有，投资者的收益为 0。因此，投资于风险资产的预期收益为 0.25(1/1.5×6－1.33)＋0.75×0＝0.67。

当 P＝1.5 时，风险资产的收益 0.67 远远大于安全资产的收益 0.17，投资者更喜欢投资于风险资产，1.5 不再是风险资产的均衡价格。造成这一结果的根本原因就是投资委托代理关系内生的风险转嫁问题。1 个单位的安全资产和 1/1.5 个单位的风险资产期末的预期价值都是 1.5。如果投资于安全资产，借款人获得 0.17，贷款人获得 1.33；如果投资于风险资产，借款人获得 0.67，而贷款人获得 0.25×1.33＋0.75×(1×1/1.5)＝1.5－0.67＝0.83。借款人的有限责任使 1.33－0.83＝0.5 的预期价值从贷款人转移给借款人，换言之，风险从借款人转移给贷款人。那么，在代理投资下风险资产的均衡价格是多少呢？由于风险资产的供给是固定的，代理投资下投资者对风险资产的偏好必然抬高风险资产的均衡价格，直到借款人对风险资产和安全资产投资的收益相等。根据 0.25(1/P×6－1.33)＋0.75×0＝1.5－1.33，可得 P＝3，即风险资产的泡沫为 3－1.5＝1.5。

假设风险资产期末价值为 9 的概率为 0.25，价值为 0 的概率为 0.75，该资产的预期价值仍然为 2.55，但是风险更大。根据 0.25(1/P×9－1.33)＋0.75×0＝1.5－1.33，可得风险资产的均衡价格为 P＝4.5。由此可见，资产风险越大，价格泡沫就越大。

11.5 金融市场泡沫的实验经济学检验

近年来，随着行为金融理论和实验经济学的兴起，越来越多的学者开始从行为金融视角出发，利用实验经济学方法，对金融市场泡沫的理论模型进行发展和完善。证券市场实验是在实验室中构造一个证券交易的环境，实验参与人作为证券交易者进行买入或者卖出证券的决策，每个交易阶段结束时交易者所持有的证券都可以获得分红。这种研究方法由 Forsythe 等人首创，继而被 Plott、Sunder、Friedman 等人进一步发展。泡沫实验的目的是检验市场泡沫是如何产生的，实验一方面可以通过设定不同实验环境来检验哪一个因素导致了市场泡沫的出现；另一方面通过控制市场的风险结构、收益结构引出交易者的

偏好，考虑是否由于交易者的心理特征及风险偏好导致了泡沫的出现。[①]

史密斯等(1988)在实验室里模拟了股票市场的交易过程，在控制市场信息结构的情形下，观察了泡沫发生的整个过程。

实验流程大致如下：实验共有 15 轮，每轮时间为 120 秒。每一轮结束时，持有的股票都会获得分红。分红有四种可能：25％的概率分红 60 元；25％的概率分红 28 元；25％的概率分红 8 元；25％的概率不分红。究竟市场实现哪种分红，在每轮结束时会宣布。在第 15 轮股票分红之后，股票的赎回价值为零，持有的现金没有利息，市场没有交易费用。实验结束时，被试的总收益等于期末持有的现金总量。

每位实验者在初期将得到 720 元的现金和 4 只股票，可以用自己的现金购买其他参与者的股票，也可以向其他参与者出售自己所拥有的股票，从而使自己的现金获得增长。实验采取双向拍卖的交易机制。

在完全理性、风险中性的假设条件下，股票的价值等于未来收益流的贴现值。

每一轮股票的期望收益为

$$EP = 60\times0.25+28\times0.25+8\times0.25+0\times0.25=24$$

那么第 1 轮交易前股票的基础价值应为

FV＝15 轮×24 元＝360(元)

第 2 轮交易前股票的基础价值应为

FV＝14 轮×24 元＝336(元)

……

第 15 轮交易前股票的基础价值应为

FV＝1 轮×24 元＝24(元)

交易结束后，股票的赎回价值为零。

这是信息完全对称的实验环境，即有关红利分布的信息对每个交易者来说都是共同知识。他们进行了多次实验，结果显示在超过一半的实验中出现了明显的泡沫现象，而且其中一次实验的参与者全部是专业的商业人士。如图 11.4 所示，实验结果显示，在 15 个交易阶段中，成交价格往往在开始时低于基础价值，然后价格逐步上升，直至出现了价格泡沫，之后泡沫破裂，成交价格在后面几个交易阶段才收敛于基础价值即理性预期的均衡价格。

为了考察信息对证券价格的影响，研究者还设置了信息不对称的实验环境。Plott 等设计了一组研究信息扩散的实验，市场中有 12 个交易者，分属于三种类型，每类交易者的红利都有两种可能性，两种可能性的概率是所有交易者都已知的。有关最终哪一种状态会实现的信息，只有 6 个交易者知道，另外 6 个交易者不知道。实验表明，开始的成交价格是偏离基础价值的，随着实验的进行，没有信息的交易者通过观察价格来推测哪一种状态会实现，对基础价值的偏离开始减少，市场信息可能从完全拥有信息的内幕人扩散到没有信息的人。

① 金雪军，杨晓兰．证券市场泡沫实验研究综述[J]．国外社会科学，2004(3)：20-25.

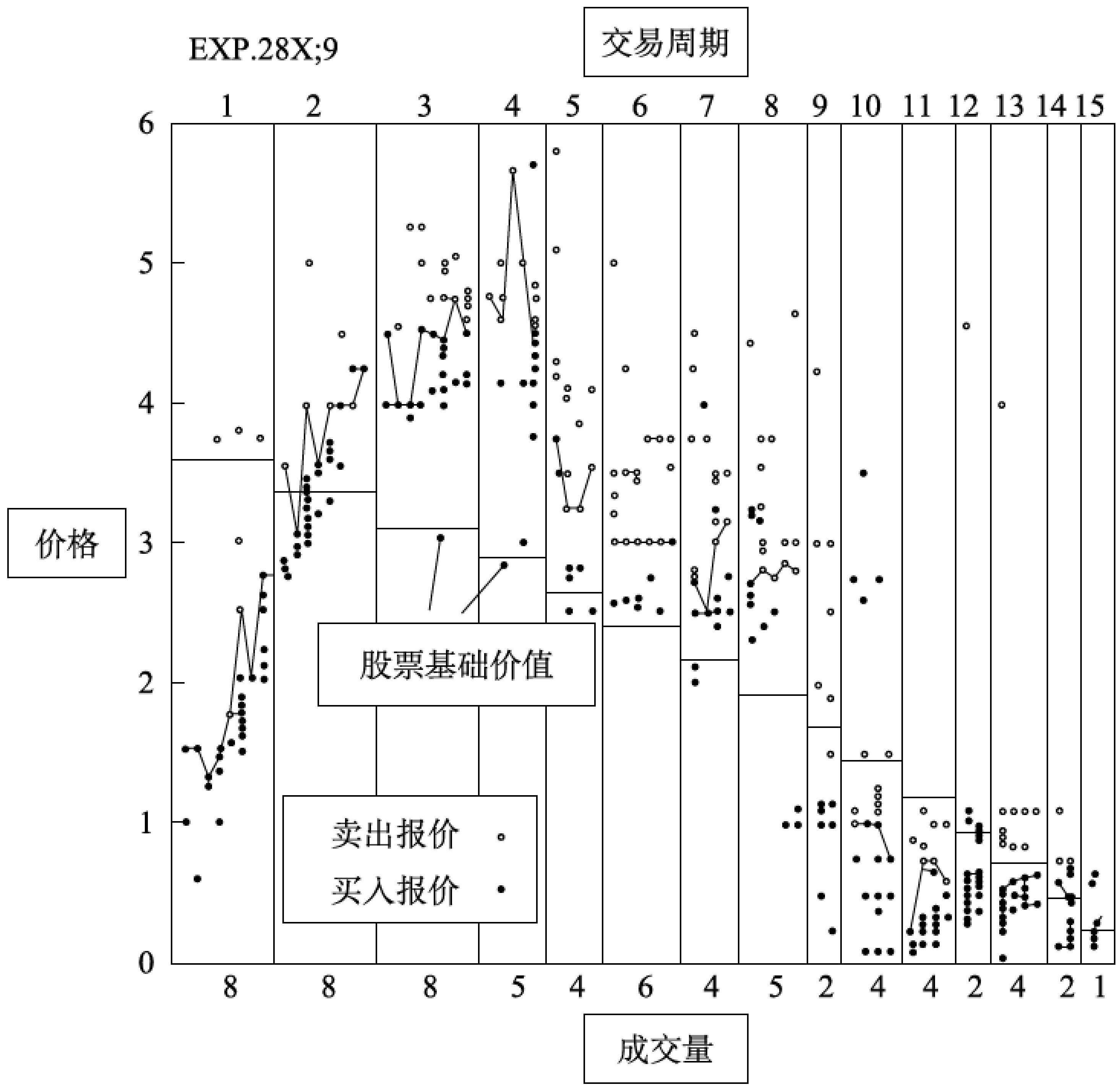


图 11.4　实验室下的股价表现

更多的学者在实验室环境下考察了不同的交易制度是否能有效抑制市场泡沫。

(1) 卖空机制:King(1993)的实验结果表明,卖空机制对泡沫的产生没有显著的影响。类似的结果也来自 Porter、史密斯,他们根据实验结果指出,卖空机制不能显著地降低泡沫量以及泡沫的持续时间。

(2) 期权交易:Forsythe 等检验了一个两阶段的资本市场(没有不确定性),发现当第一阶段的现货市场与第二阶段的期权市场相配合时,成交价格收敛于均衡价格的速度加快了。Porter 和史密斯的实验结果也表明,期权市场能够加快投资者形成共同预期的过程,从而能显著地降低市场的泡沫量,但是期权市场无法消除价格泡沫。

(3) 涨跌停制度:King 等进行了 6 次实验,实验规定价格的涨幅和跌幅不能超过单阶段期望红利的两倍。实验结果显示,价格限制制度没有阻止泡沫的出现。他们进一步指出,价格限制制度会使泡沫现象更为严重,因为价格限制让投资者感到风险降低了,从而激发他们将泡沫量提得更高,使持续时间更长。

(4) 交易费用:证券市场实验的批评者指出,实验中投资者进行交易的成本太低,仅仅是按下按钮接受或者输入报价或要价。因此,Porter 和史密斯设计了一个需要交易费用的证券市场实验,每次成交的费用是 20 美分,买方和卖方各支付 10 美分。他们的研究结果是,交易费用对泡沫的价格振幅、持续时间、股份流通量都没有明显的影响。

(5) 红利制度:这类实验主要考察红利支付频率和基础价值的变化频率是否会影响泡沫的产生。史密斯等提出了两个假设:第一,频繁地支付红利会增加泡沫产生的可能性;第二,频繁地变化基础价值也会增加泡沫产生的可能性。为了检验这两个假设,他们创造

了一种实验环境，在这个环境下证券市场的分红过程非常简单，仅在时间轴的最末端才支付红利，而且基础价值保持不变。实验结果表明，即使选取了没有经验的参与人作为交易者，在 10 次实验中，仅有 1 次出现了泡沫，其他 9 次交易价格都紧密地沿循着基础价值。史密斯等认为，他们提出的这两个假设通过了实验检验。

(6) 资本所得税：Lei 等(2002)设计了一个征收资本所得税的实验市场。他们构造的市场与其他发现泡沫及其破裂的市场是类似的，唯一不同的地方是 50%的资本所得税，也就是在每个交易阶段结束时对资本的正的利得(除了分红以外现金流的增加)征收 50%的税收。实验得出的结论是资本所得税没有消除泡沫。

课程实验指引[①]

1. 实验背景

本实验将建立一个证券市场，每个参与人作为一名投资者，在市场中进行交易。每轮实验开始前将赋予你一定数量的账面现金作为你拥有的初始资产。你的资产会由于你的投资决策相应增减，这与你的最终得到的真实收益相挂钩。具体实验将按照如下方式进行。

2. 实验框架

本次实验共 10 轮，也就是 10 个交易周期，每轮时间为 120 秒。每 12 名参与者将被分配到同一个市场中，市场之间是相互独立、互不影响的。

3. 交易资产

所有的实验者被分为 A、B 两组，两组只存在初始禀赋的差异。A 组每名投资者在初期将得到 720 元的现金和 4 只股票，B 组每名投资者在初期将得到 1 440 元的现金和 4 只股票。实验中，要交易的资产是 1 个单位的某种股票。每一轮结束时，持有的股票都会获得分红。分红有四种可能：25%的概率分红 60 元；25%的概率分红 28 元；25%的概率分红 8 元；25%的概率不分红。究竟市场实现哪种分红，在每轮结束时会宣布。在第 10 轮股票分红之后，股票的赎回价值为零，持有的现金没有利息，市场没有交易费用。

4. 交易顺序

每个市场(每个实验小组)由 12 名投资者组成。实验开始后，在每一轮中，交易者可以自行选择在任何时间内购买股票(在买入报价的框中键入买价，并单击“买入报价”)，该价格将显示在右边的卖出报价区域中，等待其他参与者以该价格卖出，或者自行在卖出报价列表中选择自己愿意支付的价格，并单击“买入”。同样地，被试可以自行选择在任何时间内卖出股票，操作与买入相反。

5. 交易机制

实验采用双向拍卖机制，即在每轮实验过程中，被试可以用自己的现金购买同一市场

---

① 实验设计参考史密斯股票泡沫实验。实验主要目的是比较市场不同的资金初始量是否会导致股东价格泡沫的差异。

上其他参与者的股票,也可以向其他参与者出售自己所拥有的股票,从而使自己的现金获得增长,每两轮之间,会有暂停页面显示该被试当前的现金与股票数目。

6. 实验报酬计算

实验报酬按照如下的方法计算。

在每轮实验结束后,电脑会根据相应的概率选择分红并向市场公布,对单个被试来说,每轮的分红收益=该轮结束时被试持有的股票数量×该轮每只股票的分红。在第10轮股票分红后,股票的赎回价值为零。

实验的理论预测结果是B组的价格泡沫高于A组,可收集实验数据对此进行检验。

本章小结

(1) 泡沫是指一种或一系列的金融资产在经历连续的涨价之后,市场价格大于实际价值的经济现象。历史上的金融泡沫时期都表现出相似的特征,如价格与价值严重背离、交易量异常持续放大、大量低认知水平投资者涌入等。一旦破灭,泡沫会造成巨大的经济损失。

(2) 现阶段对泡沫的合理测度仍然是学术界一个争议的热点,主要有基本面估计法、统计量检验法、价格序列分解法三种方法。

(3) 导致金融市场泡沫产生与发展的原因有很多,从宏观层面来讲主要有宽松的货币政策、新技术引入、代理投资、媒体偏差等因素;从微观角度,即个体行为来看,有限关注、过度乐观、羊群行为等心理偏差是引发泡沫的重要原因。

思 考 题

1. 测度金融市场泡沫有哪些方法? 除了本章提到的,你还知道哪些测度价格泡沫的方法?

2. 金融市场泡沫为什么会产生? 监管者可以采取哪些措施抑制泡沫的产生?

3. 设计一个测量股票市场泡沫的实验,可以参考本章的课程实验指引,并对其进行改进。

第 12 章

行为公司金融

引导案例：四川长虹——等离子电视的失败决策

平板电视最常见的主要有两类：液晶电视和等离子电视。提及等离子电视，不得不提及国内上规模的生产等离子电视的公司——四川长虹。

当电视从 CRT（阴极射线显像管）时代向平板电视过渡时，日系电视正如日中天，此时以东芝、索尼等企业为代表的液晶电视和以松下、日立等企业为代表的等离子电视分庭抗礼。2005 年 4 月，刚上任不久的长虹集团董事长兼总经理赵勇在董事会上提出发展等离子电视技术的想法，此时等离子技术的市场占有率在 30%左右。出于对等离子技术的看好，出身工科的赵勇认为等离子项目前景十分可观，唯技术论的他认为等离子技术能够充分创造利润和抢占市场。但是其他董事认为，相比液晶电视，等离子电视在发展上渐渐处于弱势地位，因此极力反对赵勇提出的发展计划。但对技术过度自信的赵勇认为等离子技术优势明显、投资成本较低，因此不管其他董事的意见，推动公司于 2006 年投资 25 亿美元发展等离子项目。

2007 年，四川长虹布局等离子电视的生产线，但是此时等离子电视的出货量已经远远不及液晶电视，后者在平板电视的市占率已达 90%以上，显然赵勇的计划已被市场逐步确认为非理性的投资行为。而当四川长虹于 2008 年实现等离子电视的全面量产时，液晶电视在平板电视的市场地位已无法撼动。导致等离子电视失败的最主要原因在于松下等公司对等离子技术的死守，使得其他公司无法看到掌握等离子显示技术的希望，东芝、富士通、索尼及飞利浦等相继退出了等离子市场。与此同时，液晶电视的市场参与者越来越多，技术的进步使得原先液晶电视拖尾、颜色显示等劣势得到了很好的解决。最终，一直支持等离子电视的日立于 2009 年着手退出等离子市场，领军企业松下于 2011 年开始逐步收缩并于之后停止生产等离子电视。唯一剩下的上规模的等离子电视生产公司四川长虹也逐步将重心放在液晶电视上，液晶电视出货量于 2012 年超过等离子电视。

四川长虹在等离子电视项目上的失败投资主要源于经理人赵勇的过度自信。但赵勇作出发展等离子技术的初衷对他自身而言仍是致力于股东利益最大化的考虑，而并非传统的委托代理问题。从这个案例可以看出，经理人的过度自信将会导致公司的过度投资，进而损害公司价值。传统金融以理性为假设前提，其中却并未涉及经理人的非理性偏差。本章将从投资者非理性、经理人非理性两个视角去探究其对公司融资、投资、股利决策、并购决策的影响。

12.1 行为公司金融概述

公司金融是一门试图解释金融契约与受投资者和经理人双重影响下公司的实际投融资行为的学科。传统的公司金融要求市场是有效的，参与人是理性的，市场中的投资者能够根据自己的偏好作出理性决策，能够根据标准效用与贝叶斯规则理性反映关于公司的公开信息，而经理人能够对项目的未来前景作出无偏估计，能够充分预计理性投资者的行为并作出自己理性的决策判断。但依循传统公司金融关于理性假设的前提，公司金融中存在诸多不能作出有力解释的现象，如公司的过度投资行为、盲目收购行为、偏好发放现金股利等。

事实上，已有的大量证据表明，市场并非是完全理性的。市场的参与者存在过度自信、参考点依赖、损失厌恶、从众行为等心理偏差，而这些心理偏差致使市场的参与者不能始终作出理性的决策，进而影响到公司的财务决策。行为公司金融正是基于市场参与者非完全理性的假设前提，更多地以证据驱动的行为基础代替传统金融的理性假设，将人的行为纳入公司的财务决策，结合心理学与行为认知学对投资者和公司经理人的行为偏差作出合理的解释。

如图 12.1 所示，如果按照投资者是否理性、经理人是否理性两个维度来进行分类，第一个象限就是传统公司金融模型的研究范畴，其他三个象限是行为公司金融理论的研究范畴。本章内容将梳理投资者非理性与经理人非理性分别给公司的融资行为、投资行为、股利分配及并购决策所带来的影响，并对个别重要的现象作出解释。

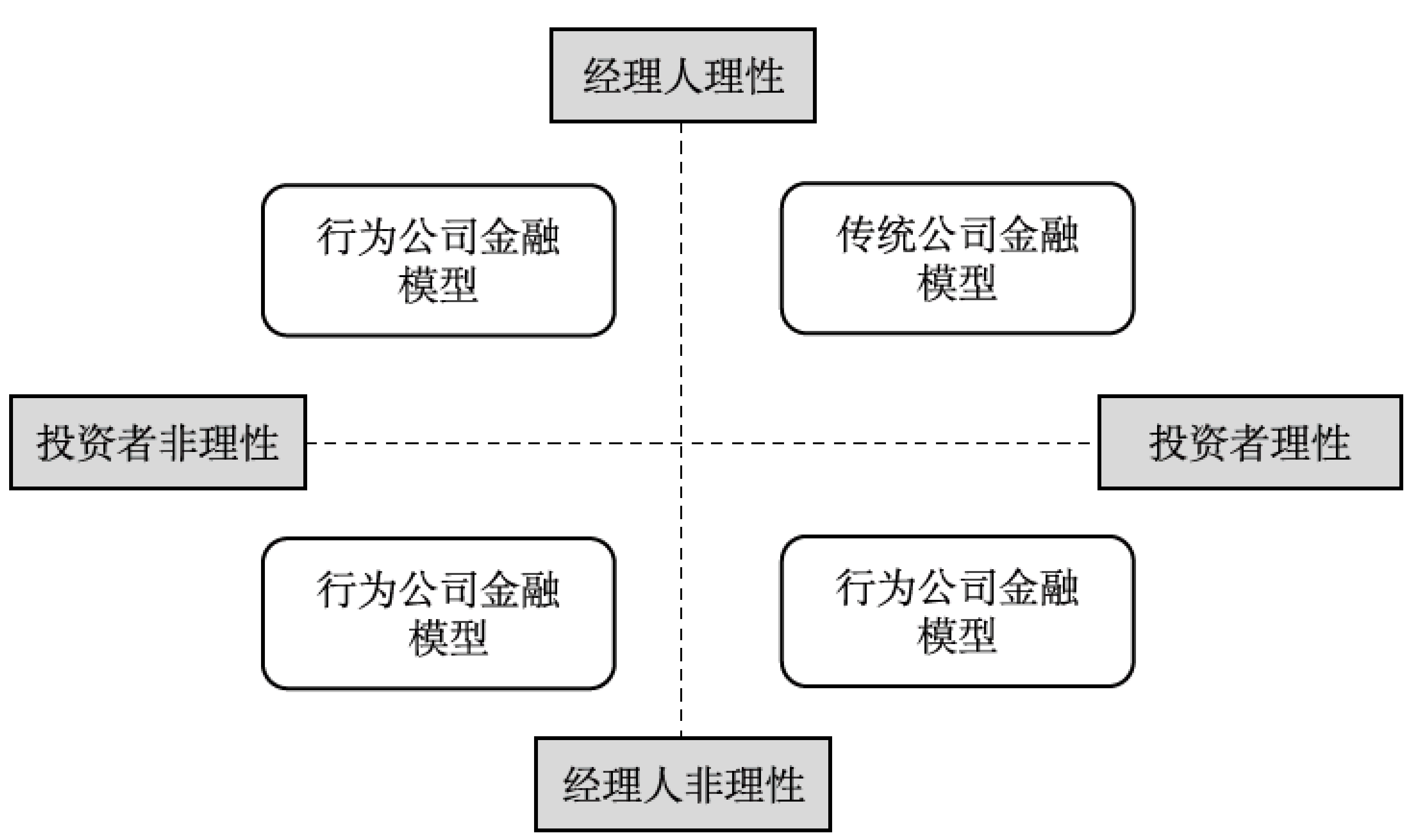


图 12.1　行为公司金融框架

注：横向坐标表示投资者是否理性，纵向坐标表示经理人是否理性。

专栏 12.1　投资者情绪与管理者过度自信

投资者情绪与管理者情绪其实存在相互影响的关系，如图 12.2 所示。

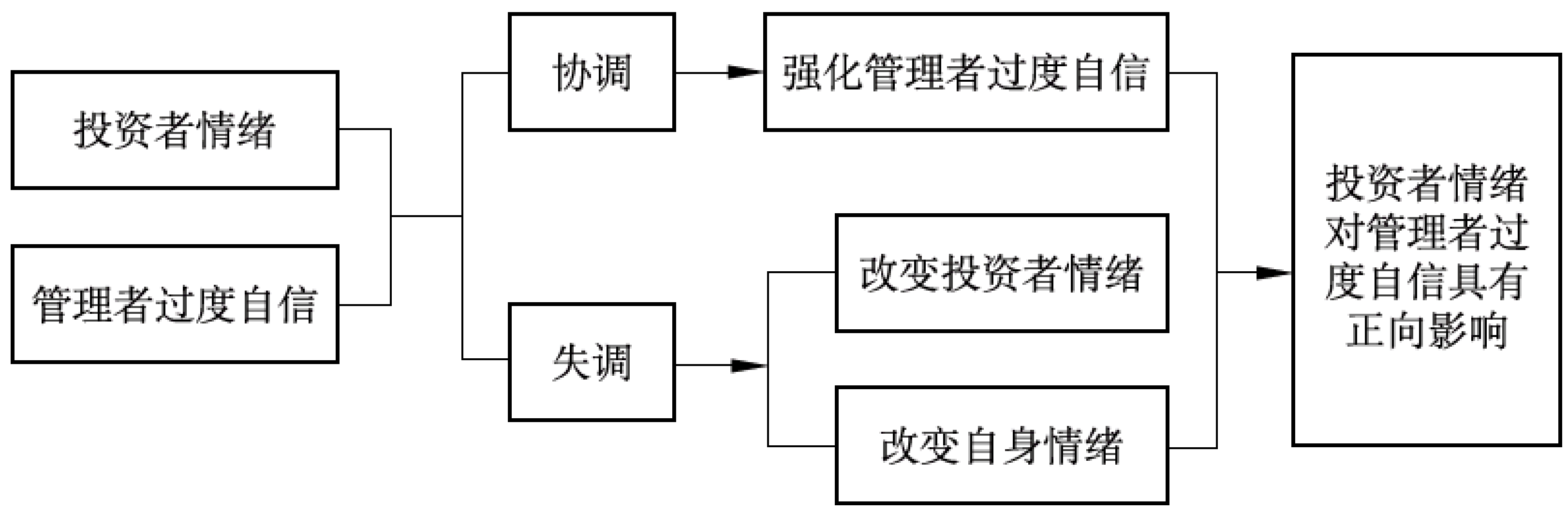


图 12.2　投资者情绪塑造管理者过度自信

首先，管理者与投资者对企业的预期收益和风险都有自己的主观认知，而管理者能够感知投资者的乐观或者悲观情绪。如果管理者与投资者对企业的预期收益和风险的主观认知一致，那么他们就处于协调关系；反之，如果投资者与管理者的认知不一致，并且被管理者所察觉，那么管理者就会产生认知失调。

如果二者处于协调关系，管理者的过度自信得到投资者的"支持"，因此将得到强化。如果二者处于失调关系，管理者便会产生减少或者消除失调的压力，管理者可以选择改变投资者情绪或者改变自身情绪。但是由于投资者情绪是投资者总体错误在资本市场上的反映，是一种系统性偏差，因此管理者是很难去改变或者诱导投资者情绪的，他们更多的是受投资者情绪的影响，也就是说高涨的投资者情绪会导致管理者的过度自信。

资料来源：花贵如，刘志远，许骞．投资者情绪、管理者乐观主义与企业投资行为[J]. 金融研究，2011(9)：178-191.

12.2　非理性对融资行为的影响

公司的融资主要分为内源融资与外源融资两种方式，不同融资方式的选择所积累的融资结果构成了公司的资本结构。经理人在融资方式的选择上考虑的核心问题是公司的融资成本。在理性假设与融资成本最低的最优资本结构的前提下，传统公司金融诞生了优序融资理论、权衡理论等。通常，传统公司金融认为，公司的融资行为与一定的投资行为相对应，即认为公司仅会在存在投资需求的情况下采取适当的融资行为。但是，现实中公司大都倾向于采用权益融资。特别是在我国股票市场，公司争先恐后 IPO 排队上市是一种常态。此外，公司的再融资需求存在用于补充流动性运营资金等原因，而并不是因为有投资项目的落地实施。

1. 投资者非理性对融资行为的影响

投资者非理性将会引起市场的非有效，将对公司的融资时机选择、回购时机选择以及资本结构三个方面产生影响。

1）企业融资的"市场时机假说"

市场上存在一定的理性投资者，但是大部分投资者的非理性占据了主导，使得市场存

在系统性的偏差，并不是完全有效的。Stein 在 1996 年基于"市场是非理性，经理人是理性"的假设，提出了著名的"市场时机假说"：当市场的乐观情绪导致公司的股价被高估时，理性的经理人应当发行更多的股票；而当悲观的情绪导致公司的股价被低估时，理性的经理人应当回购股票。

显然，股价的高低成为部分经理人是否进行权益融资的关键因素。相比于真正需要投资而进行的融资行为，部分权益融资也可以成为经理人把握市场时机为长期股东谋利的一种形式。Graham 和 Harvey(2001)对上市公司的财务总监的匿名调查显示，2/3 的人认为股票的低估或者高估的程度对于权益融资行为来说是非常重要的考虑因素。

公司利用股价高估发行新的权益，从长期来看，股价必然会回归价值中枢，这也就意味着市场投资者认购的权益回报率会很低。不管是首次公开发行还是再融资行为，只要伴随经理人的市场择机行为，相比于大小匹配的可比公司来说，这些公司的权益回报率就会较低，也从侧面印证了经理人的理性市场时机选择能力。

2）回购时机选择

在非理性的市场中，当公司的股价被低估时，公司原定的增发计划往往会推迟，直至股价上涨至一个合理的水平。更直接的是，理性的经理人会采取回购股票的措施。例如，在股灾的情况下，市场会出现大量的回购行为。Hong，Wang 和 Yu 在 2008 年指出历史的股灾时期尤其是 1987 年的 10 月和 2001 年的 9 月是公司回购潮的集中时期。

与市场融资时机选择相对，当股价被低估时，公司的经理人会认为股票足够便宜，便会采取回购行为。同时，这也是保证股东股权集中度的有效措施，如在 2015 年下半年的股灾中，部分公司由于股票过于便宜而遭到其他法人主体的"举牌"[①]，如若上市公司本身采取适当回购行为，则能够在一定程度上避免该类问题。

更为重要的一点是，从信号理论角度来看，上市公司的回购股票行为被视为公司股票被市场低估的信号。以股灾为例，因为上市公司高管本身对公司的经营要比普通投资者更了解，所以上市公司高管主动增持股票的价格往往被视为普通投资者购买上市公司股票的安全边际。

3）基于行为观的资本结构理论

一家公司的资本结构是长时间融资活动所积累的结果，通过投资项目的资金募集、并购项目的融资或者基于其他的目的，而市场时机在一定程度上是这些融资活动的决定因素。Baker 和 Wurgler 在 2002 年基于市场时机假说提出了基于行为观的资本结构理论。相比于决定资本结构的其他因素，市场时机会对资本结构产生持续的影响，资本结构是理性的经理人根据市场时机不断融资所积累的结果。

专栏 12.2　2015 年股灾中的回购行为

在熊市中，上市公司估值往往被低估，公司便会择机启动股票回购。与此同时，回购股票的行为也被视为稳定股价的有效措施之一。2015 年上半年，我国 A 股市场走出了波澜起伏的行情，上证指数于 6 月中旬创下了自 2008 年以来的新高，达到 5 178 点。但是，

① 举牌是指某法人主体连续增持某上市公司股票超 5%的行为。

此后，缺乏经济基本面支撑的股市在配资高杠杆的影响下开始经历疯狂的下跌过程，7 月初上证指数回到 3 500 点附近。为了稳定股价，各上市公司不断发布高管预计增持公告及回购股票的公告，主要目的在于顺应监管层要求稳定市场情绪和股价，如表 12.1 所示。

表 12.1 2015 年下半年股灾中部分上市公司回购行为

公司	公告时间	具体回购措施
万科 A	2015-07-06	以不超过 13.7 元/股回购不超过 100 亿元
盛和资源	2015-07-07	以不超过 19.24 元/股回购不超过 1 亿元
常宝股份	2015-07-09	以不超过 10.28 元/股回购不超过 2 亿元
海通证券	2015-07-09	回购 A 股价格不超过 18.8 元/股，回购 H 股不超过港币 17.18 港元/股，回购总金额不超过 216 亿元
海普瑞	2015-07-10	以不超过 30.17 元/股回购不超过 10 亿元
伊利股份	2015-07-10	以集合竞价方式，回购价格不超过 18.13 元/股，金额不超过 10 亿元
南京高科	2015-07-13	以不超过 16.41 元/股回购不超过 5 亿元
鲁泰 A	2015-07-14	回购 A 股价格不超过 15 元/股，回购 H 股不超过港币 11.8 港元/股，回购总金额不超过 10 亿元
徐工机械	2015-07-14	以不超过 20 元/股回购不超过 5 亿元
美的集团	2015-07-14	以不超过 48.54 元/股回购不超过 10 亿元

资料来源：Wind 公告。

2. 经理人非理性对融资行为的影响

经理人是理性的、假设主要是基于其对公司的经营信息具备更充分的认识，但部分经理人在面对市场时，有时也会受到自身心理偏差的影响，如过度自信和乐观，进而作出不理性的决策。Graham 在 1999 年便通过调查指出，财务总监存在明显的过度自信倾向，2/3 的人反映在互联网泡沫破灭前自己所在公司还认为股价被低估。

从经理人非理性角度来考虑，过度自信、乐观的经理人会表现出优序融资行为。优序融资理论是基于信息不对称的一种融资顺序理论，该理论主要认为权益融资对外更多传递了公司经营的负面信息，因此在融资选择上，经理人会逆向选择，优先考虑内源资本，其次是债务融资，最后是股权融资。其解释如下。

1）过度乐观、自信的经理人更偏好内源资本

虽然投资者是理性的，但过度自信的经理人认为市场低估了公司的基础价值，为了避免让长期投资者受到损失，经理人往往不愿意采取外部融资的方式，进而表现出内源融资优先于外源融资的偏好。

2）过度乐观、自信的经理人对债务的偏好要优先于股权

过度乐观、自信的经理人会对公司的 EBIT（息税前利润）的增长率存在较高估计，而对公司的 EBIT 波动性估计过低，因此经理人认为选择债务融资能够为长期投资者获取更大的利益。

参考点依赖也是经理人非理性的一种表现，它会对经理人的融资定价产生影响。以 IPO 价格为例，经理人往往考虑两个因素以确定价格。一是以询价区间的中间值为参考

点计算上市首日收盘价涨幅所带来的未实现的收益；二是IPO定价过低所遭受的实际损失。若收益足够大，经理人往往会忽略损失而允许IPO定价过低现象存在。从投资银行角度来看，上市首日让认购的投资者充分获利能够使得投资银行的声誉大大提高，而受参考点依赖影响的经理人也接受其制定的IPO价格，因此时常会出现IPO短期折价现象。可以说，以参考点依赖为代表的经理人非理性行为给IPO短期折价提供了一种可能的解释。

12.3 非理性对投资行为的影响

1. 投资者非理性对投资行为的影响

投资者的非理性情绪和行为会影响公司的股价，而理性的经理人能够区分股票的基础价值与市场价格，在非理性的市场中，经理人主要考虑以下三个目标：①最大化基础价值；②提高市场价格（在有效市场中该目标与第一个目标一致）；③利用错误定价进行择机与迎合。如果理性的经理人考虑得足够长远，在股价被低估时，不受融资约束的公司的投资水平不会受到影响，而受到融资约束的公司可能会减少投资。此外，不排除部分短视的经理人通过适时择机增加长期投资人的利益，这时公司的投资决策也将会发生一定变化。总体来看，历来研究主要围绕融资约束情况下的股权融资渠道理论和融资不受约束情形下的投资迎合渠道理论。

1）股权融资渠道理论

投资者非理性的情绪会引起股票错误定价，而股票的错误定价会对理性的经理人的投资决策产生影响。对于非股权依赖型的公司来说，眼光长远的经理人会通过市场时机选择进行融资，但并不会改变既有的实际投资计划。而股权依赖型的公司在受到融资约束时，经理人则不得不放弃一些具备投资价值的项目。

基于上述思想，Baker等在2003年提出了股权融资渠道理论。该理论认为，外部融资约束程度高的公司对股价波动的敏感性要高于外部融资约束程度较低的公司。也就是说，公司在面临外部融资约束时，对于具备投资价值的项目，公司也无能为力，这是公司股权依赖性所致。

2）投资迎合渠道理论

投资者情绪一方面可以通过股权融资渠道影响投资，另一方面也可以通过迎合渠道影响投资。一些学者指出，股权融资渠道理论仅适用于融资约束的大前提下，在融资不受约束时，股权融资渠道有效性受到质疑，投资迎合渠道理论由此而生。

投资迎合渠道理论认为，经理人受到短期外部治理压力的影响，会做出迎合投资者情绪的行为。投资者情绪高涨，对项目的盈利性就更为看好；如若经理人拒绝该类项目的投资，投资者很可能通过卖出股票的方式表达不满，因此投资水平会与投资者情绪呈现强相关性。

不仅是市场压力，股权激励政策也是经理人迎合行为的动因之一。经理人为了能够达到限制性股权激励的解锁条件，往往会忽视公司的长期利益，更加注重短期业绩的经理

人会存在更多的迎合行为。对于主动迎合投资者的经理人来说，研发投入相比于其他投资方式更具主观性与不确定性，因此往往成为经理人调节投资水平迎合投资者的重要工具。

2. 经理人非理性对投资行为的影响

经理人表现出经验法则偏好、过度自信及乐观、声誉维护等非理性的偏差时，会对公司的投资决策产生一定的影响。

经理人在面临资本预算方法的选取时，尽管 NPV 方法被认为是有效市场中最合适的资本预算方法，但大多数经理人可能更偏好简单原则。Graham 和 Harvey 在 2001 年做过的调查显示，超过一半的财务总监会选择不需要资本投入成本或者现金流预测的投资期回收法。

从过度自信与乐观的非理性经理人层面来看，主要从两个方面影响投资决策：一方面，过度自信会影响公司融资决策进而影响投资决策，过度自信的经理人认为公司价值被低估而不愿对外融资，极可能因融资约束而错失投资良机；另一方面，过度自信的经理人可能会高估投资项目的收益，更容易投资实际上风险较高或者净现值为负的项目，引发过度投资。

此外，经理人出于对自身声誉的维护也会对公司投资决策产生影响。例如，经理人会存在证实偏差的心理，过度自信的经理人投资了未经考证的项目造成亏损，出于对自身声誉的维护，会继续加大对亏损项目的投资，希望扭转亏损的局面。与此同时，声誉效应还可能引发经理人的羊群行为，精明的经理人具备项目的信息优势，而较差的经理人为掩饰自身能力会选择跟从精明的经理人进行投资，并且声誉越高，投资从众的倾向越严重。

12.4 非理性对股利分配的影响

股利是公司基于投融资所创造的利润给予投资者的回报。美国经济学家 Miller 和 Modigliani 在 1961 年提出的 MM 定理中指出，在完美的资本市场条件下，公司的价值仅仅与其盈利能力相关，而与收益的分配形式无关，这即是著名的“股利无关假说”。但在现实中，大部分投资者更加倾向于具有高分红的公司，对高分红的公司给予更高的溢价。Black 基于此在 1976 年提出了著名的“红利之谜”，认为相对于现金股利，资本利得征收的税率更低，因此是股利分配的更优选择，但实际上股份公司更愿意采取现金股利的方式，且投资者乐于投资这样的公司。行为公司金融从非理性的视角对红利之谜和一些现象作出了解释。

1. 投资者非理性对股利政策的影响

从投资者非理性角度出发，可以解释红利之谜的代表性观点有心理账户理论、股利迎合理论等。

1）心理账户理论

心理账户理论是指投资者在无意中将资本利得与现金股利放入了不同的心理账户，其中现金股利的心理账户属于风险厌恶的账户，而资本利得账户属于风险偏好的账户。

现实中，资本利得并不能替代现金股利。资本利得账户未来面临非常高的不确定性，即使两个账户都能完全反映公司当前的经营和股价，为防止未来的不确定导致资本利得账户的损失，损失厌恶的投资者会更倾向于现金股利账户。此外，停止现金股利支付对部分依赖现金股利用于消费的投资者来说是极大的损失。

2）股利迎合理论

股利迎合理论是指经理人为了获得短期股价提升而迎合偏好现金股利的投资者的一种理论，最早由 Baker 和 Wurgler 于 2004 年提出。该理论指出，公司支付股利是对风险的一种释放，可以减少股票持仓带来的未来不确定性，因此投资者较为偏好支付股利的公司，并给予较高的溢价。而经理人为了获得短期股价的提升，会出现股利迎合的需求，因此一些公司会倾向于发放现金股利。

在实际中，也常常出现在某一年支付股利的公司相比不支付股利的公司获得更高的市场溢价时，第二年原本不支付股利的公司便开始转向支付股利的情况，从侧面支持了股利迎合理论。

我国股票市场在 2000 年之前上市公司以国有企业为主，大部分企业上市的根本目的就是获得资金来源，加上股民早期投机心理更为严重，因此采取现金分红的公司比例较低。但在 2000—2001 年，监管当局将上市公司的再融资与现金分红挂钩的新政策出台之后，现金分红的公司比例开始迅速上升。此外，通过转增股本既可以保持决策者的自由现金流，并降低股利支付水平，同时又满足了投资者对股利分配的要求，近年来该比例逐渐上升。图 12.3 显示了我国 A 股采取不同股利分配政策的公司数量，现金分红的比例有显著上升趋势。每年在上市公司年报公布时，宣布高比例现金分红、高比例转股的上市公司，往往会受到投资者特别的关注，上市公司股利政策的选择在一定程度上迎合了投资者对现金分红、转增股本的偏好。

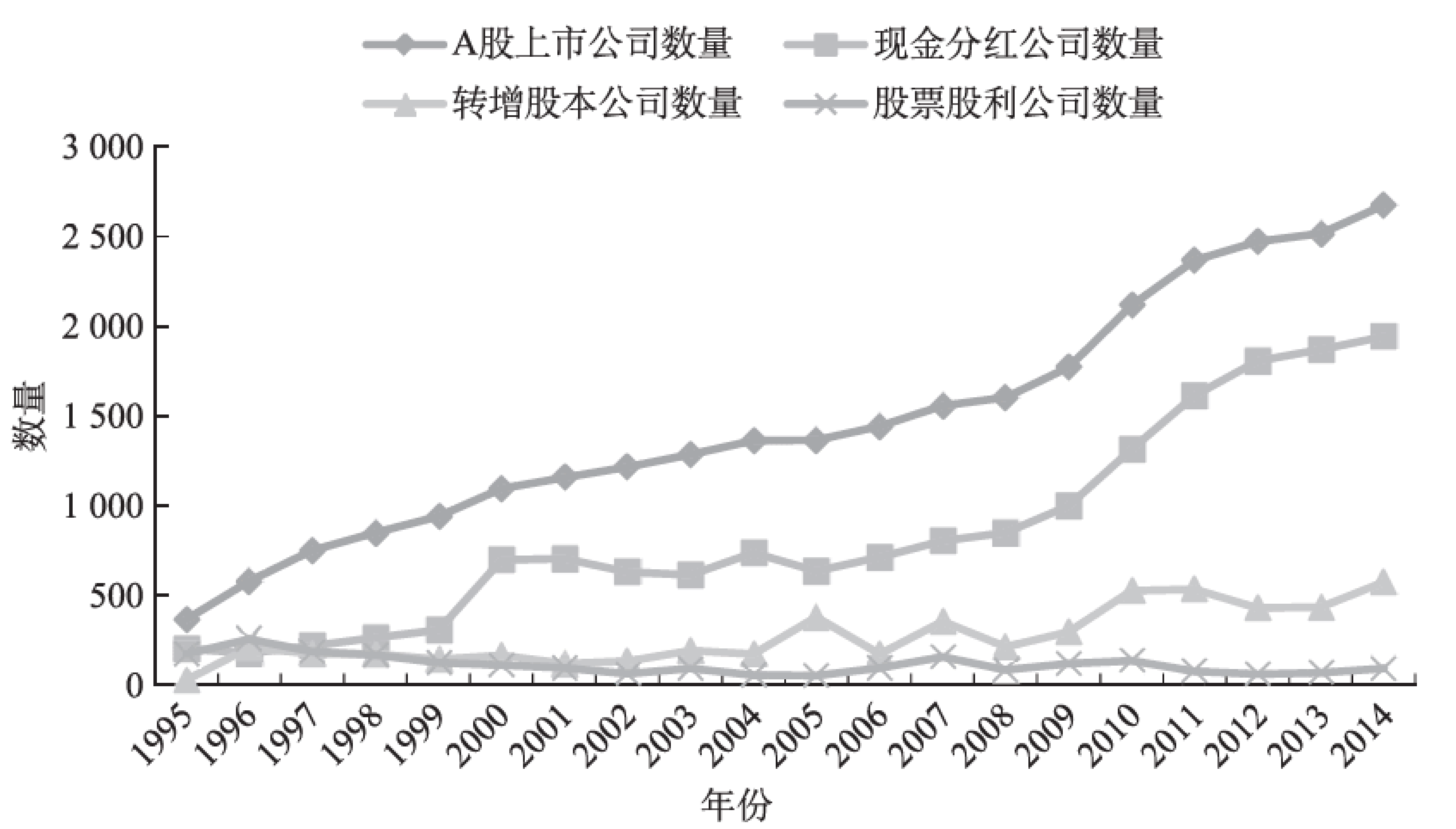


图 12.3　我国 A 股上市公司按股利分配方式统计

数据来源：国泰安数据库。

专栏 12.3 管理者的双重迎合

近年来，我国上市公司对股票送转的热情高涨，送转比例不断攀升，上市公司如何进行股票送转决策？徐龙炳和陈历轶（2018）从公司内部和外部两个方面探索了管理者在股票送转决策中的参照系选择以及迎合路径。他们发现管理者有两个进行决策的参考系：第一，公司内部可参考因素中，名义股价是管理者送转决策的参照系，送转意愿和比例跟随名义股价同向变化，为的是迎合投资者的名义价格幻觉；第二，公司外部可参考因素中，同类公司送转比例是管理者送转决策的参照系，送转意愿和比例跟随同类公司送转比例同向变化，为的是迎合投资者的参考点效应。这表明，理性管理者为了最优化股票送转决策，存在对投资者名义价格幻觉和参考点效应这两种非理性偏好的双重迎合。

资料来源：徐龙炳，陈历轶．股票送转与管理者双重迎合[J]. 金融研究，2018(5).

2. 经理人非理性对股利政策的影响

从投资者非理性的角度出发，红利之谜可以得到很好的解释。但非理性的经理人在面对理性的市场时，很容易受到经验法则的影响，或者过度看好公司未来的发展前景而做出减少分红等举措。

1）过度自信或乐观会引起分红减少

如果经理人对于未来的现金流和资产比投资者乐观，他会认为分红支出更加可持续。如果他看中未来的投资机会，那么他将会减少分红，保留内部的留存收益用于再投资，因此过度自信或乐观的经理人会主动控制分红或者减少分红。

此外，过度自信的经理人减少股利支付也符合其倾向过度投资的逻辑。过度自信的经理人对项目的前景估计和未来现金流流入都过于乐观，但在信息不对称的前提下，过度自信的经理人认为投资者低估了公司价值，因此不愿进行成本过高的外部融资，而使用内部的留存收益意味着公司将降低股利支付。

2）运用启发式或羊群行为进行股利分配

公司制定股利政策应当结合公司的长远发展考虑，合理平衡股利和留存收益再投资的比例。但现实中，经理人在制定股利政策时很容易运用启发式或者受市场中其他公司股利分配决策的影响。

公司的财务主管在制定股利政策时倾向于稳定的股利支付，并不会轻易改变股利支付的水平，主要是稳定的支付水平有利于吸引投资者。经理人在进行股利分配决策时，容易受行业优秀公司股利支付水平的影响，因此市场中公司的股利支付水平会呈现出较为集中的状态，这也可以解释为什么业绩较差的公司选择较高的股利支付水平。

12.5 非理性对并购行为的影响

公司价值及规模的扩大存在两条路径：内生增长与外延式扩张。相比来说，外延式扩张速度更加快，因此资本市场并购浪潮不断，大公司吞并小公司在全世界范围内普遍存在，随着资本逐渐积聚，并购的金额也呈现出不断递增的趋势。而之所以发生并购行为，

传统金融最主流的解释是协同效应的产生,包括经营上、财务上的协同。现实中常常出现并购发生后,并购公司在业绩上并没有呈现协同效应带来的增长,反而较之前的公司价值有所减少。此外,部分并购行为从长期来看损害了公司的价值,是无效率的并购。

1. 投资者非理性对并购行为的影响

投资者的非理性情绪及市场非有效会导致公司发生错误定价,理性的经理人会充分利用市场时机进行并购,如当市场情绪高涨导致并购公司股价被高估时,并购公司的经理人则往往采取换股并购而非现金并购的方式。同样地,当市场情绪低迷引起并购公司的股价被低估时,其往往会选择被低估的公司进行收购。而利用公司自身股价被高估进行的收购,在未来的并购收益会表现很差。

1)并购的市场时机模型

并购的市场时机模型是指因市场非有效而引起股价被高估的公司会进行资产收购,但其并不是为了协同效应,而是为长期股东保留当前股价被高估的受益。该理论最早由Shleifer 和 Vishny 于 2003 年提出,他们指出当被高估的并购公司对目标公司进行收购,尤其是带有现金资产的目标公司,并购公司能够为其股东以硬资产来充当高估股价下跌的缓冲垫。与此同时,如若同时产生协同效应,缓冲效应将具备可持续性。

此外,一旦公司股价被高估,经理人愿意出更高的收购溢价,因为任何低估值的资产的注入都有利于其自身股价下跌的部分缓冲。并且,股价越被高估,注入资产的动机越强,对标的公司的出价可能越高,并购的数额也会随之增加。

2)并购收益表现欠佳

面对投资者非理性的情绪,经理人往往会在市场情绪高涨时做出短期迎合投资者的并购行为,资本市场的投资者也偏好收购的公告,能够进一步推动估值的提升。但是,在并购行为发生后,等资本市场的情绪冷却,接下来几期公司的并购收益会逐步体现,从表现上看会远差于预期。

并购收益的较差表现也从侧面印证了并购公司在做出并购行为的择时行为。在市场情绪高涨时,并购公司会降低并购标的的筛选标准,会将一些原本并不能产生协同效应的公司纳入并购的范围,这一类并购行为在公告时可能会刺激股价继续上涨,但并不利于公司的长远发展。

专栏 12.4 失败的并购案例

兼并收购对企业来说意义重大。并购能够使企业的发展获得协同效应,扩大生产经营规模,降低成本,提高市场份额。但是并不是所有的并购都能够有这样的效果,也存在不少失败的并购案例,其中一个较为典型的例子就是 TCL。

TCL 集团股份有限公司是一家从事家电、信息、通信、电工产品研发、生产及销售的大型控股企业。但是纵观 TCL 的几次海外并购案例,可谓是"完美的失败"。从 2002 年到 2006 年,TCL 先后收购德国施耐德公司、法国汤姆逊公司的全球彩电业务以及法国阿尔卡特公司的手机业务。

TCL 本来想通过这些并购进入海外市场,利用外国企业的技术优势,进一步提高彩电

以及手机的市场占有率,但是结果却事与愿违。从 2005 年开始,一直盈利的 TCL 公司陷入亏损的泥淖。2005 年欧洲彩电业务亏损超过 8 000 万美元,TCL2005 年前三季度亏损达到 11.39 亿元人民币,净利润同比下降 379%,而造成亏损的主要原因是 2004 年的两个大型并购项目。随后,合资公司在欧洲和北美市场销售份额持续恶化,汤姆逊退出彩电销售,TCL 与汤姆逊并购失败;在 TCL 与阿尔卡特的并购中,由于剧烈的文化冲突、互信机制的缺失,业务整合失败,合资公司出现巨额亏损,销售人员离职过半,研发人员也纷纷选择离开,大批原阿尔卡特员工对并购选择了"用脚投票",这次并购也趋于失败。

管理层对并购协同效应的高估和对风险的低估是 TCL 并购失败的主要原因,管理层的过度自信在这些失败的兼并收购中发挥了很大的作用。TCL 没有意识到其收购汤姆逊彩电业务获得的 CRT 技术早已落后于主流的平板液晶电视;而 TCL 虽然收购了阿尔卡特的手机业务,却并没有获得其核心的专利技术。在并购阿尔卡特之前,TCL 并没有聘请专业的咨询机构进行分析,甚至 CEO 因为忙于汤姆逊事务而将并购移交给其他管理层。TCL 管理层仅看到了并购在战略上带来的好处,却忽视了潜在的风险,在并购前没有对这个重大的事件做好充足的调查,在并购后管理层没有采取同化策略导致了文化整合的失败。

2. 经理人非理性对并购行为的影响

传统公司金融认为,公司经理人受到委托代理关系、机会主义行为等影响,会对公司资产配置效率产生重大影响。然而,部分行为公司金融学者指出,即使不存在委托代理问题,公司经理人配置资产的效率仍会受自身心理偏差的影响,包括过度自信、过度乐观等现象。

过度自信的经理人认为自己能够在并购中将管理做好,实现协同效应,但是往往过度低估了公司在并购中文化整合的难度,最终往往成为倒霉的赢家,并购效益低下。事实上,目前并没有充分的证据表明并购为社会创造了价值。Grubb 和 Lamb 于 2000 年便指出大部分的并购行为都侵害了股东的长期利益,成功率仅仅为 20%。最主要的原因是经理人过度自信或者过度乐观,高估自己的能力,对并购的标的公司给予过高的估值,进行了不利于公司估值的并购。

更特别地,经理人有时会表现出过度自信这一心理偏差,Aktas 在 2010 年指出过度自信的经理人将更可能成为并购者。过度自信的经理人有时需要对失去自我控制进行额外的补偿,因此倾向于从事高风险活动保持自信的作风,给予目标公司更高的收购溢价。

12.6 未来展望

行为公司金融打破传统理性假设框架,从投资者非理性、经理人非理性两个视角对公司的金融决策作出解释,取得了较大的成功。但是,作为新兴的学科分支,行为公司金融仍然任重道远。

相比于传统公司金融,行为公司金融缺乏完整的理论框架与体系。传统公司金融的主流理论 MM 定理、CAPM 理论等已深入人心,而行为公司金融在模型的解释力上仍显

不足,缺乏能够真正挑战主流的传统公司金融的,真正具备代表性的理论模型。

行为公司金融从投资者非理性、经理人非理性两个视角单独分析所得出的结论往往相斥。例如,在针对融资决策的讨论中,非理性投资者所带来的影响可能是外部融资的市场时机选择,而非理性经理人所带来的结果则是内源资本的优先选择。同时将投资者非理性、经理人非理性纳入考虑,得到更清晰的结论是行为公司金融发展的未来方向之一。

延伸阅读 12.1　性别与公司金融决策:男性经理人更加过度自信

Huang 等(2012)在 Journal of Financial Economics 发表的一篇文章中指出公司金融决策与经理人的性别存在一定关联,男性经理人相比于女性经理人在公司金融决策上表现得更加过度自信。文章主要以并购决策为例,发现男性经理人相比于女性经理人会采取更多的并购和发行更多的债务。并且,从收益上看,男性经理人进行的并购收益要低于女性经理人 2%,而债务融资的回报也要低于女性经理人。此外,女性经理人对于公司盈利的预测区间相比于男性经理人更为宽泛,她们更倾向于较早去执行股票期权。上述证据均指明,在公司的金融决策中,男性经理人要比女性经理人更为过度自信。

资料来源:HUANG J,KISGEN D J. *Gender and corporate finance: are male executives overconfident relative to female executives?* [J]. *Journal of financial economics*,2012,108(3):822-839.

本章小结

(1) 行为公司金融主要基于市场参与者非完全理性的假设前提,更多地以证据驱动的行为基础代替传统金融的理性假设,将人的行为纳入公司的财务决策,结合心理学与行为认知学试图对投资者和公司经理人的行为偏差作出合理的解释。

(2) 投资者非理性将对公司的融资时机选择、回购时机选择以及资本结构三方面产生影响。经理人非理性将使公司产生优序融资行为,并影响公司的融资定价。

(3) 股权融资渠道理论和投资迎合渠道理论是解释投资者非理性对投资行为产生影响的两大主流观点。当经理人表现出经验法则偏好、过度自信及乐观、声誉维护等非理性的偏差时,会对公司的资本预算方法选择、投资水平等产生影响。

(4) 从投资者非理性角度出发,心理账户理论、股利迎合理论等是解释红利之谜的代表性观点。过度自信或乐观的经理人在面对理性的市场时,容易受到经验法则的影响,或者过度看好公司未来的发展前景而做出减少分红的举措。

(5) 在非理性的市场中,存在并购的市场时机行为,但该类并购在未来的收益会不尽如人意。即使不存在委托代理问题,公司经理人配置资产的效率仍会受自身心理偏差的影响,包括过度自信、过度乐观等现象。

思 考 题

1. 分析企业融资市场时机假说的提出背景及带来的影响。

2. 收集国内外某家上市公司回购股票的背景,并分析前后的股价表现。

3. 比较牛市、熊市上市公司资产负债率的变化,并思考可能的原因。

4. 比较分析股权融资渠道理论与投资迎合渠道理论的分析前提和逻辑。

5. 收集并分析一家上市公司历年来的股利政策,并研究股利调整前后的股价变化。

6. 查找A股市场中换股并购、现金并购的实际案例,分析并购前后特定窗口期的股价变动情况。

第四部分

行为金融学前沿



第 13 章

大数据挖掘与行为金融

引导案例:芝麻信用背后的大数据

芝麻信用是独立的第三方信用服务机构,是蚂蚁金服生态体系内的重要组成部分。芝麻信用通过大数据以及云计算等技术客观呈现个人和企业的信用状况,已经形成芝麻信用评分、芝麻信用元素表、行业关注名单、反欺诈等全产品线。人与人、人与商业之间的关系正因为信用而变得简单。

➢ 芝麻信用评估技术

芝麻信用基于阿里巴巴的电商交易数据和蚂蚁金服的互联网金融数据,并与公安网等公共机构以及合作伙伴建立数据合作,与传统征信数据不同,芝麻信用数据涵盖信用卡还款、网购、转账、理财、水电煤缴费、租房信息、住址搬迁历史、社交关系等。芝麻信用运用云计算以及大数据挖掘技术,通过机器学习模型算法,对各维度数据进行综合处理和评估,在用户身份特质、信用历史、行为偏好、履约能力、人脉五个维度客观呈现个人信用状况的综合评分。其五个维度包含的内容举例如下。

身份特质:在使用相关服务过程中留下的足够丰富和可靠的个人基本信息。评估你的身份信息是否真实全面可靠,如学历、职业资格、公安实名认证、信息稳定性等。数据来源于实名认证系统。

信用历史:过往信用账户还款记录及信用账户历史,主要包括信用卡还款历史、微贷还款记录、水电煤缴费等。具体而言,房贷、车贷、信用卡是否按时还款;预订酒店、打车是否未告知商家而擅自取消,诸如此类的网络行为都将影响个人的信用分数。数据来源于支付宝账户数据与征信系统的数据。

行为偏好:在购物、缴费、转账、理财等活动中的偏好及稳定性,如账户活跃度、消费层次、缴费层次、消费偏好等。淘宝购物行为、理财与转账行为等均将影响芝麻信用。数据来源于淘宝、支付宝等其他蚂蚁金服平台。

履约能力：稳定的经济来源和个人资产，主要考量：用户是否具有足够的财富和综合能力来偿还贷款或履行约定。支付账户余额、余额宝余额、车产信息、房产信息等数据能在一定程度上反映个人的还款能力。数据来源于支付宝、余额宝等平台。

人脉：好友的身份特征以及跟好友互动程度。主要衡量三方面的因素：关系圈、朋友圈信用水平以及社交影响力。数据来源于阿里巴巴社交系统、微博等其他网络社交平台。

➢ **芝麻信用的应用**

从信用卡、消费金融、融资租赁、抵押贷款，到酒店、租房、租车、婚恋、分类信息、学生服务、公共事业服务等，芝麻信用已经在上百个场景为用户、商户提供信用服务。

随着时代的发展，大数据正在悄悄地改变人们的生活。大数据在行为金融中又还有哪些有趣的应用呢？在本章中，将介绍大数据在投资者关注、投资者情绪、社会互动、个人信用风险评估等行为金融研究中的应用，为你揭开大数据神秘的面纱。

13.1 大数据时代的崛起①

1. 大数据的概念与特征

大数据本身是一个抽象的概念，从字面上来看，它表示数据规模的庞大，但它的实质含义并没有如此简单。针对大数据，目前存在多种不同的理解和定义。

作为各种学术和应用领域最广泛引用的一个定义，维基百科将大数据定义为无法在合理时间内通过传统的数据库工具对其内容进行抓取、管理、处理并整理成为人类所能解读的信息的数据集合。

麦肯锡在报告 *Big data：the next frontier for innovation，competition and productivity* 中指出："大数据指的是所涉及的数据集规模已经超过了传统数据库软件获取、存储、处理和分析的能力。这是一个被故意设计成主观性的定义，并且是一个关于多大的数据集才能被认为是大数据的可变定义，即并不定义大于一个特定数字的 TB（太字节）才叫大数据。因为随着技术的不断发展，符合大数据标准的数据集容量也会增长；并且定义随不同的行业也有变化，这依赖于一个特定行业通常使用何种软件以及相关的数据集有多大。因此，当下的大数据在不同行业中的范围可以从几十 TB 到几 PB（拍字节）。"

研究机构 Gartner 对大数据给出了这样的定义：大数据是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率与多样化的信息资产。

根据以上定义，可以发现大数据是一个宽泛的概念，它不仅仅是容量非常大的数据集合。从本质上来讲，它离不开维克托·迈尔-舍恩伯格和肯尼斯·库克耶提出的 4 V 特征：规模性（volume）、多样性（variety）、高速性（velocity）、价值性（value）。

第一，规模性。数据体量大是大数据区分于传统数据最显著的特征。随着信息化技术的高速发展，数据开始爆炸性增长。目前来看，大数据所处理的数量级基本上都在几十 TB 到几 PB。有关数据表明，淘宝网近 4 亿的会员每天产生的商品交易数据约 20 TB；Fa-

① 本节主要参考樊重俊，刘臣，霍良安．大数据分析与应用[M]．上海：立信会计出版社，2016．

cebook 约 10 亿的用户每天产生的日志数据超过 300 TB；百度每天通过云计算平台处理的数据超过 1.5 PB。然而，到 2012 年为止，人类生产的所有印刷材料的数据量仅为 200 PB。

第二，多样性。简单理解，多样性就是指数据的类型多样。大数据的数据类型早已不只是标准化的结构化编码数据，还包括网络日志、搜索引擎、视频、社交媒体论坛、电子邮件、图片、音频等各种复杂的非结构化数据。

第三，高速性。速度是区分大数据和传统数据的一个重要特征。在巨大数据量面前，能够做到数据的实时分析，快速地处理、传送和存储。大数据处理遵循“1 秒定律”，可从各种类型的数据中快速获得高价值的信息。

第四，价值性。大数据的商业价值高，但价值密度低。尽管拥有海量的大数据，但是发挥价值的仅是其中非常小的一部分。当然，仅仅这一小部分的数据的价值也是巨大的。

在大数据中寻找价值就好比“沙里淘金”，因此必须利用数据挖掘、机器学习、人工智能方法进行深度分析，挖掘出对未来趋势与模式分析有价值的数据，运用于交通、金融、医疗等各个领域，以创造更大的价值。

2. 大数据的产生与发展

2010 年，全球进入 Web 2.0 时代，Twitter（推特）、Facebook（脸书）、博客、微博、微信等社交网络将人类带入自媒体时代，互联网数据激增。随着新型移动设备的普及以及物联网技术的推广，更多实时获取的视频、音频、电子标签、传感器等数据也被联入互联网，数据量进一步暴增。根据美国市场调查公司 IDC 的预测，人类产生的数据量正在呈指数级增长，大约每两年就能翻一番，并且这样的增长速度将会一直持续至 2020 年。可以说，人类已经真正进入到一个数据的世界、数据的时代。

最早提出大数据时代到来的是全球知名咨询公司麦肯锡，麦肯锡称：“数据，已经渗透到当今每一个行业和业务智能领域，成为重要的生产因素。人们对于海量数据的挖掘和运用，预示着新一波生产率增长和消费者盈余浪潮的到来。”

2012 年 5 月，联合国发布了一份大数据白皮书，总结了各国政府如何利用大数据更好地服务民众，指出大数据对于联合国和各国政府来说是一个历史性的机遇，联合国还探讨了如何利用包括社交网络在内的大数据资源造福人类。

自 2012 年以来，国内互联网企业和运营商率先启动大数据技术的研发与应用，如新浪、百度、腾讯、阿里巴巴、中国移动等企业纷纷启动了大数据试点应用项目，推进大数据的发展。

大数据时代已经真正到来。大数据时代对人类的数据处理能力提出了新的挑战，也为人们获得更深刻、全面的洞察能力提供了前所未有的空间和潜力。在不久的将来，大数据将改变人们的生活、工作和思维方式，带来一场大数据的革命。

3. 大数据在金融行业中的应用

大数据的研究与应用已经在互联网、商业智能、咨询与服务以及医疗服务、零售业、金融业、通信等行业显现，并产生了巨大的社会价值和产业空间。目前，大数据应用在各行各业的发展呈现“阶梯式”格局：互联网行业是大数据应用的领跑者，金融、零售、电信、公

共管理、医疗卫生等领域的应用正在不断丰富,社会价值和经济价值得到进一步体现。

在金融行业中,大数据的应用范围比较广泛。目前,金融行业大数据的应用主要体现在精准营销、风险管控、决策支持三个方面。

1）基于大数据的精准营销

简单来说,精准营销就是在合适的时间、合适的地点,将合适的产品以合适的方式提供给合适的人。大数据能够达到这个目的。

对于如今的金融机构来说,能否利用大数据迅速有效地找到用户、理解用户、服务用户,占得先机和商机,是形成差异化竞争优势的关键所在。例如,中国招商银行利用客户刷卡、存取款、电子银行转账、微信评论等行为数据的研究,挖掘不同客户的产品偏好,每周给客户发送具有针对性的广告,提供给客户最有可能感兴趣的产品和其他信息。又如,民生银行在2014年6月上线“阿拉丁”大数据在线平台,整合了民生银行100多个业务系统源数据[包括柜员系统、实物黄金、ATM(自动取款机)、手机银行等]。依据“阿拉丁”平台和专业金融技术工具,民生银行目前能够准确计算出每位客户的利润贡献度,做到个性化定价和个性化服务,提高产品精确营销能力。

2）基于大数据的风险管控

金融的风险管控,是指依据客户消费和现金流提供信用评级或融资支持,以及利用客户社交行为记录实施信用卡反欺诈。金融风险管控对银行的业务经营至关重要,对金融秩序的稳定具有重大的积极意义。在很多情况下,金融机构可以利用大数据监测账户的行为模式,预防金融欺诈行为的发生,从而避免不必要的损失,降低经营风险。例如,“空头支票”需要钱在两个独立账户之间来回快速转账;特定形式的经济欺诈牵涉两个合谋经纪人以不断抬高的价格售出证券,直到不知情的第三方受骗购买证券,使欺诈的经纪人能够快速退出。对于诸如此类的欺诈事件,都可以借助大数据进行即时监测。

3）基于大数据的决策支持

应用大数据技术收集、处理、分析金融数据,并对数据进行挖掘提取,寻找其中有价值的信息,从而帮助公司作出及时准确的决策。

例如,阿里信贷在面对中小企业和个人贷款难问题时,基于淘宝网的企业交易数据,开发了信用评估大数据应用,能够自动分析判断是否给予企业贷款。再看一个典型的案例,美国创业公司ZestCash,主要是给信用记录差或者没有信用卡历史的客户提供个人贷款服务。它和一般银行最大的不同在于其所依赖的大数据处理和分析能力。大多数美国银行依靠FICO信用卡记录得分作出贷款与否的决策,这种个人信用评级法只考虑了15～20个变量。而ZestCash公司的信用评分模型采用了数千条信息线索,因此它能更加全面准确地评估个人的信用。

大数据不仅有助于作出正确的贷款决策,还能帮助投资者作出准确的投资决策。华尔街“德温特资本市场”公司对接Twitter,分析全球3.4亿Twitter账户的言论,进而判断社会情绪。公司根据社会情绪来决定股票的买入和卖出,获得了较高的收益率。

专栏 13.1　金融行业的大数据应用概括

Hussain和Prieto(2016)概括了金融行业三个主要的大数据应用场景,分别是市场

操纵检测、声誉风险管理以及零售经纪业务。表 13.1 所示为这三大应用场景的具体内容。

表 13.1　金融行业的大数据应用场景

名称	市场操纵检测
摘要	检测虚假谣言
简介	金融市场会受谣言的影响。很多时候谣言的目的是干扰和误导其他市场参与者。一个典型的例子就是市场试探和哄抬股价。市场试探是指非法传播上市公司的虚假信息;哄抬股价是指发布虚假的上市公司报告,吸引其他市场参与者购买股票,导致价格上涨
商业目标	识别虚假谣言,评估新信息是否可靠
名称	声誉风险管理
摘要	对银行向客户提供的咨询服务的声誉风险暴露进行评估
简介	一个负面看法可能会对银行的生存和发展产生不利的影响。违约概率(发行人信用风险)、价格波动以及在限制性市场上交换特定金融产品的难度都会增加与经纪业务和咨询服务相关的声誉与操作风险。银行和金融机构通常会向第三方提供金融产品。这意味着第三方所提供产品的性能会对银行与客户之间的关系产生实质性的影响
商业目标	监测第三方的声誉,以及识别第三方声誉毁损对银行客户之间关系的影响
名称	零售经纪业务
摘要	发现主题趋势,监测新闻事件,优化投资组合/资产分配
简介	零售经纪和市场数据行业的发展趋势是提供更多的可操作数据。他们的重点不再是基于历史数据,如关键数字或绩效数据,而是寻找具有预测性并易于理解的信号。因此,通过文本情感分析技术,获取情绪因子,已经得到了很大的应用
商业目标	自动收集和评价各种金融信息

资料来源:HUSSAIN K,PRIETO E. *Big Data in the finance and insurance sectors*[M]. New Horizons for a Data-Driven Economy. Springer,Cham,2016:209-223.

13.2　社交网络大数据挖掘

随着互联网时代的到来,科技的变革使互联网技术得到了快速发展,不仅给人们的生活带来了巨大的改变,而且加快了社会发展的步伐。根据中国互联网络信息中心(CNNIC)在京发布的第 43 次《中国互联网络发展状况统计报告》,截至 2018 年 12 月,中国互联网用户数量已达到 8.29 亿,互联网普及率达 59.6%。

国内社交网络发展迅速,出现了如豆瓣网、新浪微博、知乎等一大批知名社交网站。根据国内知名的移动大数据服务商 QuestMobile 发布的年度报告——“2016 年度 App 价值榜”,微信的月度活跃用户数已突破 8.5 亿,并且同比增长 23%,位居 APP 百强榜第一位;QQ 的月度活跃用户数超过 6.5 亿,位居第二名,但同比开始下跌;微博的月度活跃用户已达到 3.4 亿,同比增长 46%。

在这样的背景下,互联网社交网络已经成为人们获取与传递信息的重要途径,也是社交生活中最重要的组成部分。然而,传统的数据分析方法已无法处理社交网络大数据,因

此必须借助大数据挖掘技术来探索和开发社交大数据的奥秘。

数据挖掘是指从数据库的大量数据中揭示出隐含的、先前未知的并有潜在价值的信息的非平凡过程。数据挖掘是一种决策支持过程，它主要基于人工智能、机器学习、模式识别、统计学、数据库、可视化技术等，高度自动化地分析企业的数据，作出归纳性的推理，从中挖掘出潜在的模式，帮助决策者调整市场策略，减少风险，作出正确的决策。本节介绍四类主要的数据挖掘技术：网络爬虫技术、自然语言处理（natural language processing，HLP）、文本情感分析以及社会网络分析。

1. 网络爬虫技术

网络爬虫，又称网络蜘蛛，是按照一定规则能自动且快速获取网络数据的脚本程序或工具软件。网络爬虫是大数据分析的第一步。由于社交网络大数据的体量之“大”，传统的人工收集方法已经无法满足需求，而网络爬虫技术能帮助克服这一困难，实现自动且快速地获取数据，带来极大的便利。

在现有的技术条件下，有两种方式可以实现网络爬虫。

第一，使用现成的爬虫软件。这类软件已经集成了大部分爬虫的常用功能，能够解决复杂的网页结构类型，基本可以满足大部分的爬虫需求，适合新手操作。例如，集搜客（Gooseeker）爬虫软件，无须任何编程码，它就能帮你抓取网页文本、图片、视频等各种网页信息。当然，灵活性不高、速度慢也是现成爬虫软件的普遍劣势。

第二，学习编程语言，编写爬虫脚本。在这里，推荐大家运用 Python 实现爬虫。Python 是目前主流的爬虫编程语言。Python 主要有四种途径可以实现爬虫功能：Scrapy 爬虫框架、BeautifulSoup 模块库、Mechanize 模拟浏览器以及 Selenium 浏览器。

值得注意的是，通过网络爬虫，只能获取在互联网上公开并且可以访问到的网络信息，而无法获取网站后台信息（如个人用户隐私）。

2. 自然语言处理

自然语言，就是指人们日常交流使用的语言。广义的自然语言处理包含所有用计算机处理自然语言的操作，具体指词频统计、提取文本信息、文本分类、理解句子与文章含义等功能。

在学术研究中，面对海量的社交网络大数据，自然语言处理技术显得尤为重要。Pablo 等（2016）发表在国际权威期刊上的研究论文就运用了自然语言处理技术。他们基于美国的推特平台，利用自然语言处理技术筛选出提到联邦公开市场委员会会议的推特帖子，研究推特内容能否预测股票收益。研究表明，推特情绪能够预测股票收益。

具体在技术实现方面，Python 编译环境下的自然语言工具包（NLTK）可以实现文本分析的功能。自然语言工具包提供了大量的文本分析工具，包括获取语料库、文本分类和语义解释等其他处理自然语言的操作。早在 2001 年，自然语言工具包就已经被开发出来了。之后，该模块不断发展壮大，如今已成为主流的自然语言处理工具。

3. 文本情感分析

文本情感分析，又称意见挖掘，是指通过计算技术对文本的主客观性、观点、情绪、极性的挖掘和分析，对文本的情感倾向作出分类判断。文本情感分析是自然语言理解领域

的重要研究分支。

行为金融领域主要将文本情感分析技术应用于投资者情绪的研究中。其中最具代表性的文献就是 Antweiler 和 Frank(2004)的研究。他们获取了雅虎财经的 150 万条帖子，首先随机抽取了 1 000 条帖子进行人工分类，作为训练数据集，然后采用朴素贝叶斯算法将剩余的测试数据集自动分成看空、看多、中立三类，由此构建投资者看涨指数与意见分歧指数，他们发现前一天的发帖量与第二天的收益率呈显著的负相关关系，同时投资者意见分歧指数与交易量呈显著的正相关关系。

目前，主要有基于情感词典和基于机器学习这两种技术方法对文本进行情感分析研究。

基于情感词典的分析方法。其基本思路是首先对具有情感倾向性的代表性词汇进行情感标注(一般分为"积极""中立"和"消极"三类情感)，构建一个情感词典。然后运用已经分类的情感词典对句子中的词语进行分类，进一步采用统计技术来评估文本内容的情感。

基于机器学习的方法。其基本思想是运用机器学习模型对文本情感进行自动分类。在机器学习中比较成熟且常用的分类方法有支持向量机(support vector machine)、朴素贝叶斯(Naive Bayes)和最大熵模型(maximum entropy)等。在机器学习中，数据被分为训练集与测试集。一般将文本的情感分为三类："积极""消极"与"中立"。需要事先对训练集进行人工分类，在训练集数据的基础上选择机器学习算法进行训练，再将模型应用到测试集中，最终实现测试集文本情绪的自动分类。

4. 社会网络分析

社会网络分析是一种独特的社会研究方法，用于研究社会网络中行动者之间的关系，这里的行动者包括人、社区、群体、组织、国家等。它是借助图论和矩阵等数学理论对社会关系结构及其属性加以分析的一套定量分析方法，主要从规模、紧密程度、聚类分析以及节点中心性四个方面来分析社会网络。

社会网络由节点和连线组成，通常人们用节点表示社会的行动者，而用节点之间的连线表示行动者之间的相互关系。基于节点和连线的社会网络表现了行动者在社会环境中的相互作用，这种模式也反映了社会结构。

社会网络分析方法受到越来越多学者的青睐与关注，目前在社会学领域发展出了社会学习、社会传染、社会互动、网络博弈等若干个重要的研究方向。

近年来，经济学家也开始关注社会网络问题，将社会网络分析方法应用到经济学研究中，尤其在行为金融领域，社会网络为投资者行为分析提供了一般性的分析框架，深化了对群体行为的认知，推进了行为金融理论的发展。

1) 社会网络规模

社会网络规模是指社会网络中所有行动者的数目，一般情况下，社会网络规模越大，社会网络的复杂性也越高。现实社会网络的规模一般不是很大，基本上均小于 1 000。而基于互联网社交平台的社会网络规模庞大，复杂性较高，规模可能在几千或者几万的数量级。

2）社会网络紧密程度

社会网络紧密程度是指社会网络中各节点之间关系的复杂程度。社会网络的紧密程度越高，网络对其行动者的影响就越大。网络密度与平均路径长度这两个指标可以衡量社会网络的紧密程度。社会网络密度越大，平均路径长度越短，表明社会网络的紧密程度越高。

3）社会网络聚类分析

物以类聚，人以群分。聚类分析就是研究社会网络是否存在聚类效应，即同一个社交网络中的行动者之间的关系是否紧密。用聚类系数来衡量一个社会网络的聚类效应，它是所有节点聚类系数的平均值。通过聚类分析，可以发现社会网络的子群体（社区），该子群体中的联系比其他行动者之间的联系更加紧密。

4）社会网络节点中心性

节点中心性是指节点在网络中的重要程度，衡量该节点在网络中是否重要或者居于怎样的地位。它是判断节点在网络中重要性的指标。节点中心性指标有四种：点度中心性、接近中心性/亲密中心性、中介中心性/中间中心性以及特征向量中心性。节点中心性最主要应用于专家的发现和意见领袖的识别。意见领袖在网络中具有重要的地位和影响力，对社会网络中的信息传播和行为传染起着重要的作用。

专栏 13.2　朴素贝叶斯算法简介

朴素贝叶斯算法是应用最为广泛的文本分类模型之一。学过概率论的同学一定都知道贝叶斯定理：

$$P(B_i \mid A) = \frac{P(B_i)P(A \mid B_i)}{\sum_{j=1}^{n} P(B_j)P(A \mid B_j)}$$

看上去贝叶斯公式只是把 B 的后验概率转换成了 A 的后验概率＋B 的边缘概率的组合表达形式，因为很多现实问题中 $P(B|A)$ 或 $P(A \cap B)$ 很难直接观测，但是 $P(A|B)$ 和 $P(B)$ 却很容易测得，利用贝叶斯公式可以方便我们计算很多实际的概率问题。

朴素贝叶斯算法正是基于贝叶斯定理与特征条件独立假设的条件概率模型。它的核心思想是用新的观测值去更新条件概率。通过贝叶斯公式和特征条件独立假设，我们不难推导出：

$$P(T_1 \mid W_N) = P(T_1) * \exp\left[\sum_{i=1}^{N} \ln\left(\frac{P(W_i \mid T_1)}{P(W_i \mid T_2)}\right)\right]$$

其中，N 为词条总数；W_i 为第 i 个词条；T_1、T_2 为两种不同的类别。

13.3　基于大数据挖掘的投资者关注研究

1. 投资者关注的含义

关注是心理学上的一个概念。德国著名心理学家 Wundt 首先在心理学中引入了关注的研究。在注意力成为心理学研究的初期，学者们主要关注点在于人们是否能同时关注

两个或两个以上的事物,也就是关注能不能被分开,这实际上是有限关注理论的前声。卡尼曼(1973)从心理学角度提出了有限关注理论,指出对一事物的关注必然以牺牲对另一事物的注意为代价。他认为人们是能同时做两件事情的,但是如果需要的关注度资源超过所能提供的,那么做第二件事情的时候,就会使第一件事情效率下降。

在经济学研究中,Hirshleifer(2003)首次对关注给予了经济学解释,他认为关注主要包括两个内在过程:其一是编码过程,即对外部环境刺激进行内部化处理,如投资者看到了公司的盈余公告;其二是对意识思维进行处理的过程,如投资者对公司盈余公告进行分析。

投资者关注是如何影响股票市场的呢?在股票市场中,传统金融理论认为,新的信息可以及时反映到资产价格中。而实际上,投资者并不能够对信息进行及时准确的解读,这种现象无法被有效市场假说理论所解释。例如,Ball 和 Brown(1958)首次发现了盈余公告后持续漂移现象。行为金融理论认为,投资者关注的不足导致对盈余信息解读的不完全是出现市场异象的原因。具体而言,人们在处理多类信息时,存在着精力和能力的局限性,导致投资者无法快速充分地理解新的信息,进而导致资产定价的偏差。这在行为金融中被称为有限关注理论。有限关注理论认为,投资者关注作为投资者特定的心理现象,会对投资者决策产生重大的影响,继而影响股票的价格和成交量,造成股票市场短期的剧烈波动。

2. 投资者关注的传统度量方式

公众注意力衡量了一只股票受投资者关注的广泛程度,受到越多的投资者关注,那么股票的公众注意力就越高。投资者关注理论的研究属于新兴领域,在该领域的研究中,难题之一便是对投资者关注度的度量问题。目前学术界尚未有统一的标准,多数学者通过各种间接指标(金融资产的交易行为和价格特征)来衡量投资者的关注度。传统的投资者关注度代理变量主要有以下五种。

第一,异常收益率。异常收益率是指某种证券的实际收益率与市场预期收益率之间的差值。正常情况下,异常收益率接近于零。当股票的异常收益率显著为正或负时,认为投资者对该股票产生了更多的关注。异常收益率越高,投资者关注就越高。国内学者许柳英和陈启欢(2005)将异常收益率作为投资者关注的代理变量,研究了投资者关注对投资者投资决策的影响。

第二,异常交易量。与异常收益率类似,异常交易量是指某种证券的实际交易量与市场预期交易量之间的差值。异常高的交易量,代表了更多投资者的参与,也就是投资者的高关注度。因此异常交易量也可以作为投资者关注的一个代理变量。

第三,股票换手率。换手率指在一定时间内市场中股票转手买卖的频率,代表了市场的活跃程度。目前来看,国内外多数学者将换手率作为投资者关注的衡量指标。Loh(2009)以分析师荐股公告作为研究事件,以前 3 个月的平均换手率作为投资者关注度的代理指标,发现投资者关注减轻了市场对分析师荐股公告的反应不足。权小锋(2010)等采用换手率指标来衡量投资者关注,实证检验了投资者关注度对盈余公告效应的影响,结果表明,投资者关注能够减弱盈余公告效应。

第四，股票涨停板。Seasholes 等(2007)认为，涨停板事件能够吸引投资者的关注。他们根据股票涨停当日以及下一日的账户交易数据设计了一个买卖不平衡变量，将其作为投资者关注的代理变量。他们认为，如果涨停板事件增加了投资者关注，那么买卖不平衡变量值就是正的。研究结果表明，股票涨停板事件带来了新的投资者，增加了投资者关注，因此涨停板事件可以作为投资者关注的代理变量。

第五，广告投入。Grullon 等(2004)认为，公司的广告投入费用也能作为投资者关注的代理变量。广告，作为公司的宣传手段，在拓宽产品销售渠道的同时，使公司通过电视、杂志、广播、网络等方式被投资者所知晓，提高公司知名度，进而增加投资者关注度。他们的研究表明，广告投入越多，股票的投资者越多，股票流动性也越好。

虽然传统的投资者关注指标能在一定程度上量化投资者的关注度，但是也存在一定的局限性。这些指标只是金融资产交易行为的具体表现，因此并不能直接反映投资者的关注心理，往往会存在很大的偏误。

3. 数据挖掘的应用

通过上一节的学习可以知道，投资者关注的传统度量指标均是间接指标，并不能直接反映公众注意力的分布和强弱，需要寻找更加直接准确的度量方式来衡量投资者关注。所幸的是，网络大数据为对投资者关注的直接度量提供了数据支持，可以利用大数据挖掘技术，直观地观测到投资者对公司的关注情况，进而获取投资者关注指标。

1）基于媒体报道的投资者关注

在股票市场中，媒体作为最主要的信息传播媒介，在公司与投资者之间搭建起了一座信息沟通的桥梁。当然，在传播信息的同时，媒体报道也能引发投资者的关注。因此，媒体报道也可以作为投资者关注度的代理变量，包括媒体报道有无、媒体报道数量等指标。

国内学者饶育蕾等(2010)基于网络媒体数据，利用数据挖掘技术考察了媒体关注度和股票收益之间的关系。她认为，互联网媒体已经成为我国投资者获取信息的主要渠道，因此网络媒体的新闻报道往往能够引导投资者的关注。饶育蕾运用网络编程检索技术，自动获取新浪网特定时段内各家上市公司的新闻报道数目，剔除新闻数目为零的公司，构建了一个公众注意力指标：$AT_{i,t}=\dfrac{N_t-\text{rank}_{i,t}}{N_t}$，其中 rank 表示上市公司 i 在时段 t 期的新闻报道数目在所有上市公司中的排名，N 表示 t 期所有上市公司的总数。研究发现，媒体对于上市公司的关注度越高，则下个月的股票收益越低。饶育蕾运用“过度关注弱势”理论来解释这种现象，她认为过度关注导致过度交易与过度反应，进而造成收益率的长期反转。

2）基于搜索引擎的投资者关注

不难发现，媒体关注度指标存在些许缺点：第一，媒体报道的数量与投资者关注可能并不是高度相关的，即不能保证大部分投资者会阅读媒体新闻；第二，不同新闻媒体对投资者的影响力不尽相同，不能一概而论。

随着 21 世纪互联网技术的飞速发展，搜索引擎也逐渐成为获取信息的重要渠道。投资者通过搜索引擎可以快速查询自己关注的信息，如上市公司介绍、公司公告、媒体报道

等。普通网民对上市公司的一次搜索，正代表了投资者对上市公司的一次关注。因此，可以通过搜索引擎构建投资者关注指标。这类指标也正好弥补了媒体关注度的缺陷。

互联网搜索引擎公司为获取搜索量数据提供了方便快捷的工具。2006 年 5 月，Google 公司正式推出 Google Trends，普通网民可以随时查询某一关键词在某一时段内的搜索量以及变化趋势。国外 Da(2011)等最先采用 Google 搜索指数作为投资者关注的代理变量。经过实证研究，他们发现 Google 搜索指数与交易量、异常收益率、媒体报道等间接的代理变量存在一定的相关性，并且它更能体现中小投资者的关注度。同时，他们发现基于 Google 的投资者关注度与股票收益呈正相关关系。2007 年 11 月，国内百度公司的百度搜索指数(Baidu index)正式上线。自此，百度搜索指数被广泛应用到国内的投资者关注研究中。俞庆进和张兵(2012)将百度搜索指数作为投资者关注指标，研究了投资者有限关注与股票收益之间的关系。他们发现，投资者关注在短期内会促进股票价格的上涨，而随着时间的推移会在长期内发生反转。

专栏 13.3 投资者本地偏好与股票收益

俗话说："老乡见老乡，两眼泪汪汪。"从古至今，家乡情结一直是人类情感世界中最丰富、最珍贵的一个部分。一个有趣的现象是，研究者们竟然在股票市场中也发现了这种家乡情结。也就是说，投资者更愿意关注和购买自己家乡的股票。

董大勇和肖作平在 2011 年观察到了投资者本地偏好现象。他们利用 A 股中注册地位于四川和辽宁的企业与东方财富发帖者的 IP 地址，发现了两个有趣的现象：第一，比起外省的股票，四川和辽宁的网民更喜欢交流本省的股票，并且如果公司的业务本地化水平越高，他们被本省网民讨论的频率也越高；第二，投资者本地偏好对股票价格具有正向的影响。

我们在 2016 年也对投资者本地偏好现象进行了研究。我们将样本量扩大到了全国，以投资者在东方财富网股吧对创业板上市公司发表的 90 多万条帖子为研究对象，通过 IP 地址识别，构建本地关注指标；并利用计算机文本挖掘技术，提取网络发帖所体现的情绪倾向，构建投资者情绪指标。实证结果显示，本地关注对股票收益率的影响取决于投资者情绪，当投资者持积极情绪时，本地关注对股票收益率有显著的正向影响；当投资者持消极情绪时，该影响显著为负。这个特点在上市公司样本数量较多的北京、广东、江苏、上海、浙江这五个区域都存在，并与整体样本的特征基本一致。

资料来源：杨晓兰，沈翰彬，祝宇．本地偏好、投资者情绪与股票收益率：来自网络论坛的经验证据[J]. 金融研究，2016(12)：143-158；董大勇，肖作平．证券信息交流家乡偏误及其对股票价格的影响：来自股票论坛的证据[J]. 管理世界，2011(1)：52-61.

13.4 基于大数据挖掘的社会舆情研究

1. 社会舆情的含义

"舆"字，在我国古代被理解为车，"舆人"就是造车工的意思。到了汉代，所有与统治

阶级相对应的普通民众都可以被称为“舆人”。因此，自从唐代出现“舆情”二字后，其含义并没有随着时代的发展而发生变化。在《现代汉语词典》(第7版)中，“舆情”的意思是“公众的意见和态度”。总的来说，社会舆情，是指社会民众以媒介为载体对社会公共事务所持有的情绪、意愿、态度和意见的总和。社会公共事务既包括社会事件、社会热点问题、社会冲突、社会活动，也包括公众人物的言行举止等。

现如今，网络舆情是社会舆情的一种主要表现形式。网络舆情是以网络这种媒介为传播载体的一种社会舆情，具体指公众通过互联网公开表达的对某一社会问题或社会现象的具有一定影响力和倾向性的共同意见。

网络舆情在其传播方式和效果上有很多自身特点。中国舆情网认为网络舆情的特点具体表现在以下几个方面。

(1) 开放性。互联网给予了所有人发表意见和参政议政的权利，使每个人成为信息的发布者。同时，由于大部分互联网平台不需要实名制，网民可以客观真实地表达自己的意见和观点。

(2) 多元性。网络舆情的主题极为宽泛，话题的确定往往是自发的、随意的。从舆情主体的范围来看，网民分布于社会各阶层和各个领域；从舆情的话题来看，涉及政治、经济、文化、军事、外交以及社会生活的各个方面；从舆情来源上看，网民可以在不受任何干扰的情况下预先写好言论，随时在网上发布，发表后的言论可以被任意评论和转载。

(3) 突发性。互联网能对突发事件作出快速反应，凭借星星之火就能形成燎原之势，能迅速聚集网民的看法形成公众意见，造成强大的舆情声势。

(4) 偏差性。由于网民的文化水平和生活状况不尽相同，导致部分言论缺乏理性判断，容易情绪化，甚至很多网民将互联网当成宣泄压力的场所，因此极易出现片面、偏激的言论。

毋庸置疑，作为一种社会的共同价值取向，社会舆情必然会对股票市场产生影响。其对股票市场的影响机制与投资者情绪类似。社会舆情会影响投资者的行为决策，进而对股票市场产生影响。

2. 社会舆情的传统度量方式以及数据挖掘的应用

在网络没有那么发达的年代，量化社会舆情是一项非常困难的任务。研究者只能通过问卷调查或者访问调查的方式，获得个人对社会公共事务的意见和态度，将所有意见聚集起来，进行量化社会舆情。然而，这种研究方法的弊端显而易见，不仅耗时耗力，而且研究的数据样本量远远不够。

现如今，网络舆情在社会舆情中占有主导地位。所幸的是，随着数据挖掘技术的发展，网络舆情的量化极大地降低了收集数据的成本，并且克服了问卷调查样本量不足的缺陷。肖欣荣等(2016)曾做过一个简单的研究。从2014年10月起，微博大V但斌在微博上多次谈论中国平安股票，其更是在2014年12月26日发布一则强烈看好中国平安的微博。该微博被转发422次、评论107条、点赞113次。随后在2015年前5个月内，中国平安股票价格上涨幅度超过50%。肖欣荣的研究发现，在但斌于2014年10月持续谈论中国平安后，百度搜索关键词“但斌新浪博客”和“中国平安”的移动趋势指数之间的相关度

持续走高，并与同期中国平安股票的走势高度相关。国外学者 Johan(2011)等进行了一项有关公众情绪和股票收益的研究。他们基于国外社交平台 Twitter，运用谷歌情感分析工具(Opinion Finder)和六维情感分析工具 GPOMS 来分析文本情绪，得到积极、消极、平静、警惕、确信、重要、友好和快乐七个情绪指标，实证结果表明公众情绪对道琼斯指数变化有显著的积极影响。他们还发现，神经网络模型引入公众情绪后，股市收益的预测准确率得到了显著提高。

专栏 13.4 新浪财经—数据平台：微博舆情与市场热点

2015 年 6 月中旬，新浪财经推出了基于沪深交易所 Level2 行情的新一代行情系统，除了 Level2 十档行情、逐笔交易数据和资金流向等常见数据外，还包含新浪财经独家的微博大数据股票分析和市场热点追踪分析功能。

微博大数据股票分析通过对新浪微博每天数以亿计的博文进行计算，分析每一条微博中提到的股票以及博文作者或转发者对某只股票是看多还是看空，构建个股的微博舆情指标，并以直观的方式展示给用户。

同样，市场热点分析追踪功能利用新浪微博的博文数据，基于关键词对每一条博文进行热门主题的分类，根据微博的数量、转发、评论等指标，挖掘最新的市场热点。

13.5 基于大数据挖掘的社会互动研究

1. 社会互动的含义

什么是社会互动？社会科学领域对这个问题还没有形成统一的答案。社会互动一词最早来源于社会学。早在 1908 年，德国著名社会学家齐美尔就从互动的角度研究社会学，认为互动形式是社会互动的核心。巴克(1984)认为社会互动是一方或多方的反应，取决于或依赖于另一方所说所做的程度，社会情景则随着这种程度而变化。近年来，经济学家也开始关注社会互动的研究，Durlauf 和 Loannides(2010)从经济学视角出发，将社会互动定义为“个体之间的相互依赖性，在这些相互依赖性之下，一个社会经济行为人的偏好、信念及其面临的预算约束受到其他行为人的特征和选择的直接影响”。

总结借鉴前人的研究：社会互动是个体、群体之间沟通信息、交流感受的过程，并且这个过程会对互动双方产生不同程度的影响。

国外学者 Manski 认为，在新经济社会学理论体系下，个人的行为分析既要考虑一般经济行为理论中的偏好、风险因素，又要考虑个人所在的社会群体对他们的影响。也就是说，应该把社会互动理论纳入一般经济行为理论的体系。

同样在股票市场中，投资者的个人决策也将受到个人所在的社会群体的影响。对于投资者而言，信息是投资者决策的关键因素，而社会互动在传递信息方面发挥着重要的作用。由于投资者自身精力有限，他们无法关注市场上的所有信息，只能借助他们的社交网络，如朋友、亲戚、邻居等，以获取相关信息，进而作出决策。

社会互动是投资者行为的重要影响因素，而投资者行为又决定了股票市场的表现。

由此可见，社会互动会影响投资者的信息感知和解读，进而影响股票市场。

2. 社会互动的传统度量方式

自从经济学研究将社会互动概念纳入其分析框架，越来越多的实证研究探讨了社会互动与经济行为之间的关系。由于社会互动的原始数据难以获取、指标难以构建等特点，过去的学者主要通过设计实验或问卷调查的方式来衡量社会互动。

1）设计实验

设计实验，包括实验经济学和仿真模拟两种模式，是以实验经济学理论为基础，尽可能地模拟真实的社会互动情景，进而研究社会互动与经济行为之间的关系。在变量比较复杂的情况下，实验通常能够精准地控制相关变量，形成控制组和实验组，进而比较两组的不同结果，得出结论。

目前来看，该方法在国外比较流行。Duflo、Salganik、Tae-Seok 等很多国外学者运用实验的方法展开社会互动的研究。Duflo 和 Saez(2003)在一家公司进行了一项有趣的实地实验，首先向一部分员工发送邀请函，告知他们参加退休计划信息交流会将会获得金钱奖励，由此设置实验组和对照组，观察实验组和对照组中的员工是否会加入 TDA 退休计划。实验结果表明，同伴效应是影响是否加入 TDA 退休计划的重要因素；同时，社会互动是个人获取信息的主要渠道。也就是说，个人决策会受到社会互动的显著影响。

2）问卷调查

除了设计实验之外，国内外大量学者还通过问卷调查的方式量化社会互动指标。他们大多采用春节拜访、邻居关系、参加集体活动、人际关系等指标作为社会互动的代理变量。

国内学者李涛在这方面做了一些研究，他有两篇有关社会互动与金融投资的代表性论文。在《社会互动、信任与股市参与》这篇论文中，李涛调查了广东省 7 个城市共 2 059 名 18 岁以上的、有广东省户口并居住满 1 年的城市居民，根据问卷中的相关问题，分别构建了两组社会互动指标、两组居民参与股市指标以及一个居民信任程度指标。两组社会互动指标分别是春节拜年总人数和人际交往程度的主观评价。前者是居民社会关系的客观事实，后者是居民对自己的主观评价。两组居民参与股市指标分别是居民是否拥有股票和居民金融资产中股票资产的重要性。前者代表居民参与股市的可能性，后者代表居民参与股市的程度。李涛发现，社会互动和信任都推动了居民参与股市。在另一篇论文《社会互动和投资选择》中，李涛认为，内生互动对个体投资选择的影响反映在获得信息、交流感受、社会规范三个方面。因此，他利用 2005 年中国 12 个城市的投资者行为调查数据，分别构建了获得信息、交流感受、社会规范和情景互动四个变量，研究社会互动对投资者投资决策的影响。李涛的研究证明：第一，积极的社会互动推动了个体当前和未来期望的广泛的投资项目参与；第二，社会互动对个体当前和未来期望的投资参与的积极作用主要是通过内生互动的社会规范机制来实现的；第三，内生互动带来的感受交流和信息获得分别推动了个体当前对保险金和未来期望对债券的投资，而情景互动降低了个体未来购买股票的期望。

毋庸置疑，设计实验和问卷调查这两种方法有一定的局限性。一方面，变量值与实际

情况可能存在偏差。这两种方法是基于一部分人或者小规模群体获取的指标，无法代表整个社会群体。另一方面，它们都需要消耗大量的人力、物力，成本太高。因此，需要寻找更加便捷有效的方式来衡量社会互动。

3. 数据挖掘的应用

随着互联网技术的快速发展，特别是网络社交平台的大规模普及，网络社交平台已逐步取代传统的社交网络，成为投资者传递信息的主要途径。人们不再拘泥于通过传统社交网络获取信息的方式，而更多的是通过网络互动更加快速便捷地获取信息。投资者不仅能在网络上接收信息，还可以发起信息，主动与他人交流，进而形成大规模的基于互联网的社交网络。基于互联网背景下的社会互动是以互联网为平台的一种互动方式，称为网络互动。

网络大数据为社会互动的度量提供了有利的条件。截至目前，国内外已有大量学者利用大数据挖掘技术来衡量社会互动。编者也曾做过一项社会互动的研究，基于新浪财经博客平台，运用网络爬虫、文本情感分析以及社会网络分析等技术，探讨社会互动与股票市场之间的关系。首先，通过网络爬虫技术抓取了新浪财经博客 2013 年 12 月至 2015 年 11 月的 14 多万条博文，包括博文内容、博主 ID、发表时间、阅读、转发、喜欢、收藏等数据；同时抓取了新浪博客平台所有博主的相互关注信息。其次，利用机器学习算法进行文本情感分析，构建投资者情绪指标。再次，运用 Gephi 软件构建所有博客博主的社会网络，计算每一位博主的节点中心性。最后，通过以上所有数据，可以构建四类社会互动变量：博主互动诉求、博主影响力、互动覆盖范围和互动程度。表 13.2 展示了四类社会互动变量的含义。

表 13.2 四个社会互动变量的含义

社会互动变量	变量定义	变量符号
博主互动诉求	每周博客数量	IA1
	每周博客数量 * 情绪指数	IA2
博主影响力	每周博主中心性	II1
	每周博主中心性 * 情绪指数	II2
互动覆盖范围	每周博客阅读量	IC1
	每周博客阅读量 * 情绪指数	IC2
互动程度	每周博客评论/转发/收藏/点赞量	IL1
	每周博客评论/转发/收藏/点赞量 * 情绪指数	IL2

资料来源：杨晓兰，王伟超，高媚，《股市政策对股票市场的影响——基于投资者社会互动的视角》，工作论文，2019 年

经过模型检验，发现社会互动对股市收益率和波动率具有积极作用，社会互动会放大股市的政策效应。

专栏 13.5 通过社会网络识别意见领袖

在数据获取步骤中，我们已经获取了博主之间的相互关注信息。然后我们使用 Gephi

软件绘制新浪博客的社会网络结构。图 13.1 展示了新浪博客 1 277 位博主相互关注的社会网络。在社会网络中,节点代表博主个体,节点之间的连线代表博主之间的关系。节点的颜色越深,节点的形状越大,代表节点的特征向量中心性越大,也就是越重要。通过统计,我们找到了排名前九位的意见领袖,分别是但斌、巴曙松、侯宁、李大霄、管清友、花荣、蓝鲸财经记者工作平台、洪榕、水皮。

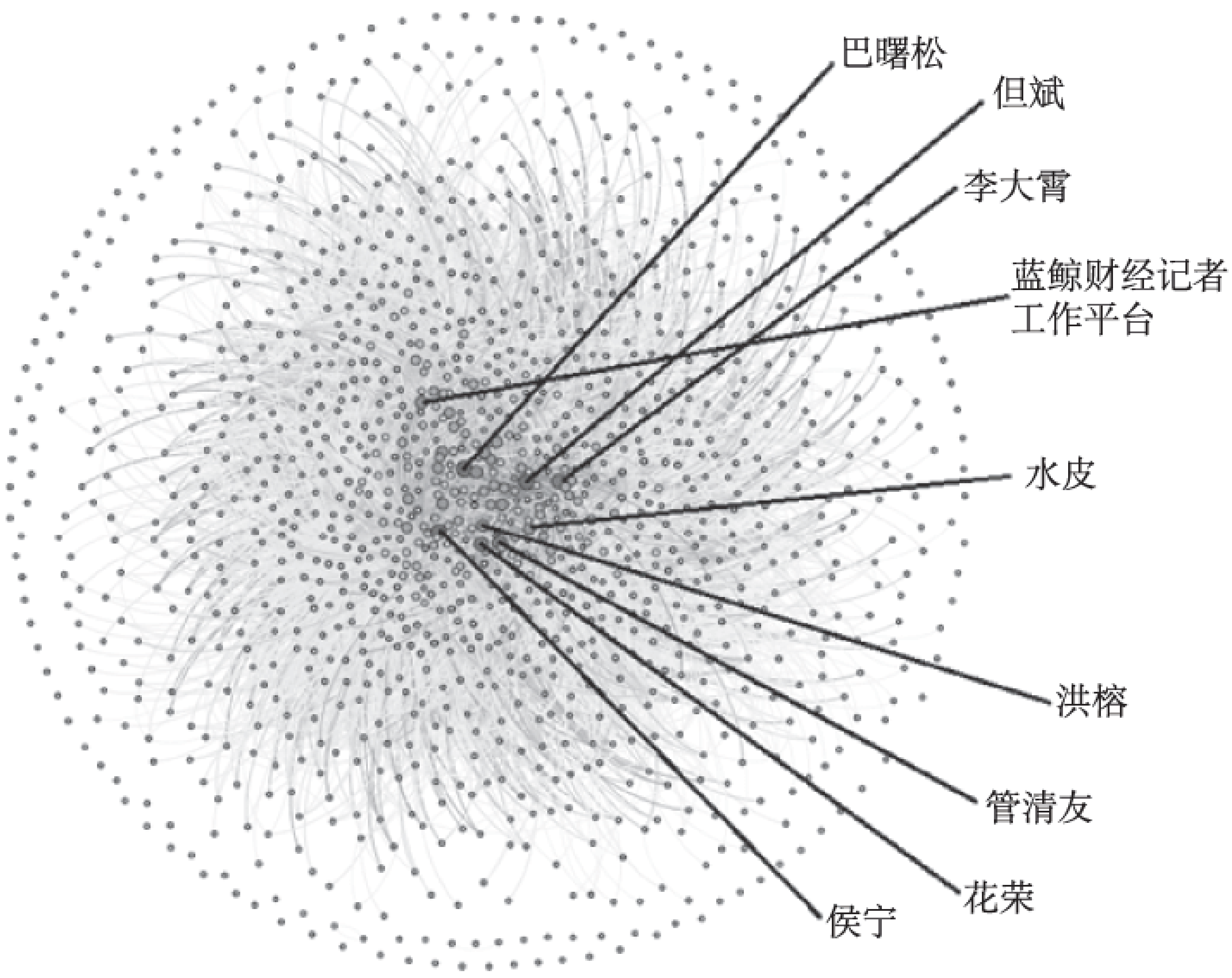


图 13.1　新浪博客的社会网络结构

资料来源:杨晓兰,王伟超,高媚,《股市政策对股票市场的影响——基于投资者社会互动的视角》,工作论文,2019 年

13.6　基于个人行为数据挖掘的信用风险评估

1. 个人信用风险评估概述[①]

什么是信用风险?《巴塞尔协议》是如此定义的:信用风险是交易对象无力履约的风险,即债务人未能在债务合同约定期限内归还其欠款,从而对经济主体未来活动产生损失的风险。顾名思义,个人信用风险是信用风险的一种,有别于一般企业的信用风险,具体指个人作为债务人未能在债务合同约定期内归还其欠款的风险。

信用风险的成因又是什么呢? 简单来说,个人信用风险主要是由于借款人和贷款人之间的信息不对称,由此引发道德风险而产生的。其中,交易双方之间的信息不对称是诱因,而道德风险是信用风险形成的重要因素。在当前个人征信系统不完善、数据缺乏的情况下,贷款人往往无法对借款人的信用、风险类型和偿还资金的概率作出可靠的判断。同时,由于这种明显的信息不对称,道德风险就成了信用风险形成的重要因素。借款人可能

① 本部分主要参考 MBA 智库百科。

会故意隐瞒个人信息，或者将贷款用于风险较高的投资，进而造成信用风险。

目前存在的传统的个人信用风险评估方法主要有 5C 要素分析法、财务比率综合分析法以及多变量信用风险判别模型等。

1）5C 要素分析法

5C 要素分析法是金融机构对客户做信用风险分析时所采用的专家分析法之一。它主要集中在借款人的道德品质（character）、还款能力（capacity）、资本实力（capital）、担保（collateral）和经营环境条件（condition）五个方面进行全面的定性分析以判别借款人的还款意愿和还款能力。有些银行将其归纳为“5W”要素，即借款人（who）、借款用途（why）、还款期限（when）、担保物（what）及如何还款（how）。还有的银行将其归纳为“5P”要素，即个人因素（personal）、借款目的（purpose）、偿还（payment）、保障（protec-tion）和前景（perspective）。无论是“5C”“5W”或是“5P”要素法，在内容上都大同小异，它们的共同之处都是将每一要素逐一进行评分，使信用数量化，从而确定其信用等级以作为其是否贷款、贷款标准的确定和随后贷款跟踪监测期间的政策调整依据。

2）财务比率综合分析法

由于信用危机往往是由财务危机引致而使银行和投资者面临巨大的信用风险，及早发现和找出一些预警财务趋向恶化的特征财务指标，可判断借款或证券发行人的财务状况，从而确定其信用等级，为信贷和投资提供依据。基于这一动机，金融机构通常将信用风险的测度转化为企业财务状况的衡量问题。因此，一系列财务比率分析方法也应运而生。财务比率综合分析法就是将各项财务分析指标作为一个整体，系统、全面、综合地对企业财务状况和经营情况进行剖析、解释与评价。这类方法的主要代表有杜邦财务分析体系和沃尔比重评分法，前者是以净值报酬率为龙头，以资产净利润率为核心，重点揭示企业获利能力及其前因后果；而沃尔比重法是将选定的七项财务比率分别给定各自的分数比重，通过与标准比率（行业平均比率）进行比较，确定各项指标的得分及总体指标的累计分数，从而得出企业财务状况的综合评价，继而确定其信用等级。

3）多变量信用风险判别模型

多变量信用风险判别模型是以特征财务比率为解释变量，运用数量统计方法推导而建立起的标准模型。运用此模型预测某种性质事件发生的可能性，及早发现信用危机信号，使经营者能够在危机出现的萌芽阶段采取有效措施改善企业经营，防范危机；使投资者和债权人可依据这种信号及时转移投资、管理应收账款及作出信贷决策。目前国际上这类模型的应用是最有效的，也是国际金融业和学术界视为主流的方法。概括起来有线性概率模型、Logit 模型、Probit 模型和判别分析模型。其中判别分析模型最受青睐，Logit 模型次之。

2. 个人行为数据挖掘的应用

由于信息不对称和道德风险的存在，依赖于财务指标的传统的信用风险评估模型在很多情况下也无法准确评估个人的信用风险。自 21 世纪以来，互联网的快速普及，越来越多的学者开始研究基于个人行为数据的信用风险评估模型，越来越多的互联网公司也开始利用互联网行为数据开展互联网征信业务。

由于个人行为数据的稀缺性，国内只有少数学者对此展开了研究。国内学者张万军(2015)构建了基于大数据的个人信用风险评估模型。他认为互联网的个人行为数据具有以下三个特点：第一，数据的稀疏性较强；第二，数据覆盖面广；第三，单变量的风险区分能力弱。因此他通过将用户在互联网上网购、交易、社交等平台的商誉和行为数据进行整合与分析，将分散在不同网络平台和信贷机构的局部信息加工融合成为具有完整视图效果的全局信息。深度挖掘互联网大数据信息，将用户商誉和行为信息转化为信贷评级依据，提出大数据环境下的个人信用风险评估模型——CreditNet 研究框架。

CreditNet 模型的用户属性基础数据依托国内某著名的大数据服务平台，该平台已经与 1 500 家商家进行业务合作，其中线上与线下各种零售商和品牌商户等大约 700 家，线上媒体和社区等约有 800 家，与其开展合作的各类金融机构达到 70 家左右；积累了 4 亿～5 亿用户的消费数据、阅读习惯、社交记录和资产信息等方面的数据。

CreditNet 模型分为三个阶段：针对第一阶段，张万军将用户画像的概念引入个人信用风险评估模型中，从身份信息核查标签、稳定性信息标签、金融申请信息标签、重要资产信息标签、商品消费信息标签、媒体阅览信息标签六大维度近 1 000 个子项构建用户信用画像，有效解决了大数据的稀疏性以及风险区分能力弱的缺点。针对第二阶段，张万军结合随机森林模型和 Logistic 模型，构建 RF-L 模型，为个人信用风险评估奠定模型基础。针对第三阶段，他运用 AdaBoost 集成学习算法，改进优化了第二阶段的模型，增强了 CreditNet 模型的评估效果。

同时，国内业界也对基于大数据的个人信用风险评估进行了研究和实践。2015 年，8 家机构成为中国首批商业征信机构，拿到个人征信牌照，这其中包括芝麻信用管理有限公司和腾讯征信有限公司。近年来，阿里巴巴和腾讯公司都积累了大量的互联网用户行为数据，这为征信业务的开展提供了得天独厚的条件。据芝麻信用发布的报告显示，截至 2017 年 3 月，芝麻信用免押模式已经深入八大行业，包括酒店、租房、民宿、租车、共享单车、医疗、便民服务以及农业设备租赁，同时全国 381 个城市已接入信用免押服务，有近 2 000 万人使用过信用免押服务，中国的城市正在向“信用城市”升级。

专栏 13.6　芝麻信用与 FICO 评分模型的区别

表 13.3 展示了芝麻信用评分模型与 FICO 评分模型的区别。

表 13.3　芝麻信用评分模型与 FICO 评分模型的区别

芝麻信用评分模型	FICO 评分模型
用户信用历史	人口统计学信息
行为偏好	历史贷款还款信息
履约能力	历史金融交易信息
身份特质	人民银行征信信息
人脉关系	

FICO 评分是美国 Fair Isaac 公司开发的信用评分系统，也是目前美国应用得最广泛的一种。FICO 评分系统得出的信用分数范围在 300～850 分，分数越高，表明客户的信用

风险越小，它采集客户的人口统计学信息、历史贷款还款信息、历史金融交易信息、人民银行征信信息等，通过逻辑回归模型计算客户的还款能力，预测客户在未来一年违约的概率。

(1) 人口统计学信息：如客户年龄、家庭结构、住房情况、工作类别及时间等。

(2) 历史贷款还款信息：过去 6 个月或 12 个月的付款方式、逾期次数等。

(3) 历史金融交易信息：过去 6 个月或 12 个月的平均月交易笔数、金额等。

(4) 银行征信信息：过去 12 个月中新开的账户总数、所有账户的总额度、账户是否逾期等。

不同的是，阿里巴巴推出的芝麻信用分则是以大数据分析技术为基础，采集多元化数据，包括传统的金融类交易、还款数据，第三方的非金融行为数据，以及互联网、移动网络和社交网络数据等，根据用户信用历史、行为偏好、履约能力、身份特质、人脉关系五个维度，帮助贷款方从多个方面考察个体的还款能力、还款意愿，作出合理、全面的信用评分。

本章小结

(1) 大数据是指无法在合理时间内通过传统的数据库工具对其内容进行抓取、管理、处理并整理成为人类所能解读的信息的数据集合。

(2) 数据挖掘是指从数据库的大量数据中揭示出隐含的、先前未知的并有潜在价值的信息的非平凡过程。数据挖掘是一种决策支持过程，它主要基于人工智能、机器学习、模式识别、统计学、数据库、可视化技术等，高度自动化地分析企业的数据，作出归纳性的推理，从中挖掘出潜在的模式，帮助决策者调整市场策略，减少风险，作出正确的决策。

(3) 投资者关注是指投资者对特定事件曝光所产生的关注行为。

(4) 社会舆情是指社会民众以媒介为载体对社会公共事务所持有的情绪、意愿、态度和意见的总和。

(5) 社会互动是指个体之间的相互依赖性，在这些相互依赖性之下，一个社会经济行为人的偏好、信念及其面临的预算约束受到其他行为人的特征和选择的直接影响。

(6) 个人信用风险是信用风险的一种，有别于一般企业的信用风险，具体指个人作为债务人未能在债务合同约定期内归还其欠款的风险。

思考题

1. 大数据的概念、特点是什么？请举例说明大数据在金融行业中的应用。

2. 大数据挖掘技术具体有哪些？行为金融研究是如何应用这些技术的，试着举例加以说明。

3. 投资者关注、社会舆情以及社会互动的含义是什么？如何运用数据挖掘技术来衡量这些指标？请详细说明。

4. 相对于传统的信用风险评估模型，芝麻信用有哪些优势？请具体说明。

第 14 章

互联网金融中的行为金融

引导案例：比特币的前生今世

2017 年 5 月 12 日，一个名叫“WannaCry”的勒索病毒通过 Windows 系统的漏洞感染了全球近百个国家的大量计算机，成为迄今为止最大规模的一次勒索病毒攻击事件。无论是校园网用户还是机场、银行、加油站、医院等企业单位，都无一幸免。中毒用户的文件全部被加密，并需支付高额赎金才能得以恢复。

在这场大规模的病毒攻击事件中，黑客声称只接受比特币作为付款方式。这让人不禁疑惑：作为一种虚拟的电子货币，比特币是如何跨越国界，成为一种独特的支付货币？

2009 年，比特币由一个名为中本聪的人设计产生，是一种虚拟的电子货币。接入网络的电脑下载比特币程序，并奉献出一定的 CPU（中央处理器）运算能力来不断地计算由该程序产生的数学题，最先解出题目的电脑用户将获得一定数量的比特币作为奖励。在该规则下，所有的比特币将在 2140 年被挖掘完毕，达到 2 100 万个的最终值。

相比于传统的流通货币，比特币有着明显的特色。

全球性：比特币广泛分布于网络环境中，没有中央银行，任意一台接入互联网的电脑都可以进行管理。因此，比特币的流通范围遍布全球，自由而灵活。

私有性：用户通过私钥来管理比特币，而私钥可能被隔离保存在任何介质上，这保证了比特币的私有性与安全性。

低成本：一方面，比特币的交易费用很低，免税，免监管；另一方面，支付比特币没有额度与手续限制，隐藏成本较低。

跨平台：用户在任何平台都能够运行程序并挖掘比特币，并比较不同硬件的运算能力。

比特币从产生至今，经历了一个泡沫发生发展的过程。在创立之初，比特币的价格仅为 5 美分。随着比特币的理念逐渐地得到传播并逐步为人们所接受，比特币的价值开始上涨。在对比特币未来走势乐观估计的影响下，越来越多的人追加了资金的投入。不断

上涨的价格增加了人们的投资热情，即使有部分投资者认为价格已过高，仍会有更多的投资者进入并推动价格上涨。由此，比特币的价格水涨船高，泡沫最终形成。2017 年 6 月 13 日，比特币的价格首次突破了 3 000 美元的大关，达到 3 012.05 美元，是 2016 年年末价格的两倍以上，是其创立之初价格的 6 万多倍。

在疯狂的比特币市场上，个体投资者的心理偏差演变成了群体性的偏差，并使市场产生了异常的表现。随着投资者的持续蜂拥而入，人们早已不考虑比特币的内在价值是多少，而是以一种投资套利的心态，希望能够在狂热的市场中赚取高额收益，并让更晚进入的投资者“接盘”，从而加速了比特币价格的泡沫化。

可以说，比特币的价格泡沫是一次披上互联网外衣的“郁金香泡沫”，生动展现了互联网金融(internet finance)领域存在的各类市场异象。

本章将从互联网金融概述出发，介绍互联网金融的主要模式及发展现状，并根据现有的研究成果，分众筹(crowd funding)、P2P 网贷(peer-to-peer lending)、亲社会网贷(pro-social lending)与其他互联网金融模式四块，介绍不同互联网金融模式下的行为金融现象。在互联网金融领域，羊群行为、个人特征与信息不对称是最显著的三大行为金融特征。因此，本章在介绍不同互联网金融模式下的行为金融时，都会在不同程度上涉及上述三大特征，同时介绍各类互联网金融模式下特有的行为金融现象。

14.1 互联网金融概述

根据中国人民银行等十部门发布的《关于促进互联网金融健康发展的指导意见》，互联网金融是指传统金融机构与互联网企业利用互联网技术和信息通信技术实现资金融通、支付、投资和信息中介服务的新型金融业务模式。互联网金融以互联网工具为依托，积极扮演好资金融通、支付和信息中介服务的角色。

目前的“互联网＋金融”格局，主要由传统金融机构与非金融机构组成。传统的金融机构主要是将传统的金融业务同互联网相结合，并衍生出各类 APP 软件；非金融机构则基于互联网技术进行各类金融运作，如 P2P 网络借贷、众筹、手机理财 APP 及第三方支付平台等。

1. 主要模式

综合来看，目前互联网金融的发展模式有众筹、P2P 网贷、亲社会网贷、第三方支付、数字货币、大数据金融、信息化金融机构、互联网金融门户(the internet financial portal)八大类。

众筹，意指大众筹资或群众筹资，指的是创业企业、艺术家或个人，通过创意或项目展示，来向网友募集项目资金的融资模式。

P2P 网贷，即点对点信贷，指的是基于互联网平台将众多的借、贷双方进行匹配，既帮助借款人以较为合理的利率筹集资金，也帮助贷款人合理控制贷款风险。

亲社会网贷，即有利于社会的网络贷款行为，指的是出于各种公益目的，如帮助社会弱势群体，协助发展中国家发展经济等，并借助互联网平台所构建的网络贷款融资模式。

第三方支付，是指非金融机构作为收、付款人的支付中介所提供的网络支付、预付卡、

银行卡收单以及中国人民银行确定的其他支付服务。

数字货币指的是可用于真实的商品与服务交易的虚拟货币，比特币是其中的一种代表币种。

大数据金融是基于客户的海量数据进行全方位的分析，获取客户交易与消费的准确信息，进而预测客户行为，最终服务于金融机构及金融服务平台。

信息化金融机构，指的是通过互联网技术的运用，对银行、证券及保险等金融机构实现信息化的统一经营与管理。

互联网金融门户，指的是利用互联网进行金融产品销售以及提供第三方服务的网络平台。

基于上述八大发展模式，目前互联网金融表现出低成本、高效率、广覆盖、快发展、弱管理、高风险等显著特点。

2. 发展现状

近年来，在国家政策扶持、经济发展、互联网迅速普及的大背景下，我国的互联网金融产业获得了极大的发展，成就显著。其中，众筹、P2P 网贷、第三方支付、数字货币与互联网金融门户的发展成果最为突出。

1）众筹

在众筹领域，来自社会各界的资金正被广泛地应用于灾害重建、竞选造势、创业筹资、艺术设计、软件开发及科学研究等各行各业，从学术界到文艺界，从草根创业到国家大事，都有着众筹的影子。如图 14.1 所示，近年来，我国众筹市场交易规模增长迅速，已从 2012 年的人民币 1.9 亿元增长到 2016 年的人民币 175 亿元。截至 2016 年年底，全国众筹平台数量共计 511 家，其中正常运营的众筹平台有 415 家，平台下线或众筹业务下架的有 89 家，转型平台共计 7 家。[①] 据网贷之家资料显示，淘宝众筹、京东众筹、众筹网、苏宁众筹和人人天使分列 2016 年国内众筹平台的前五位。

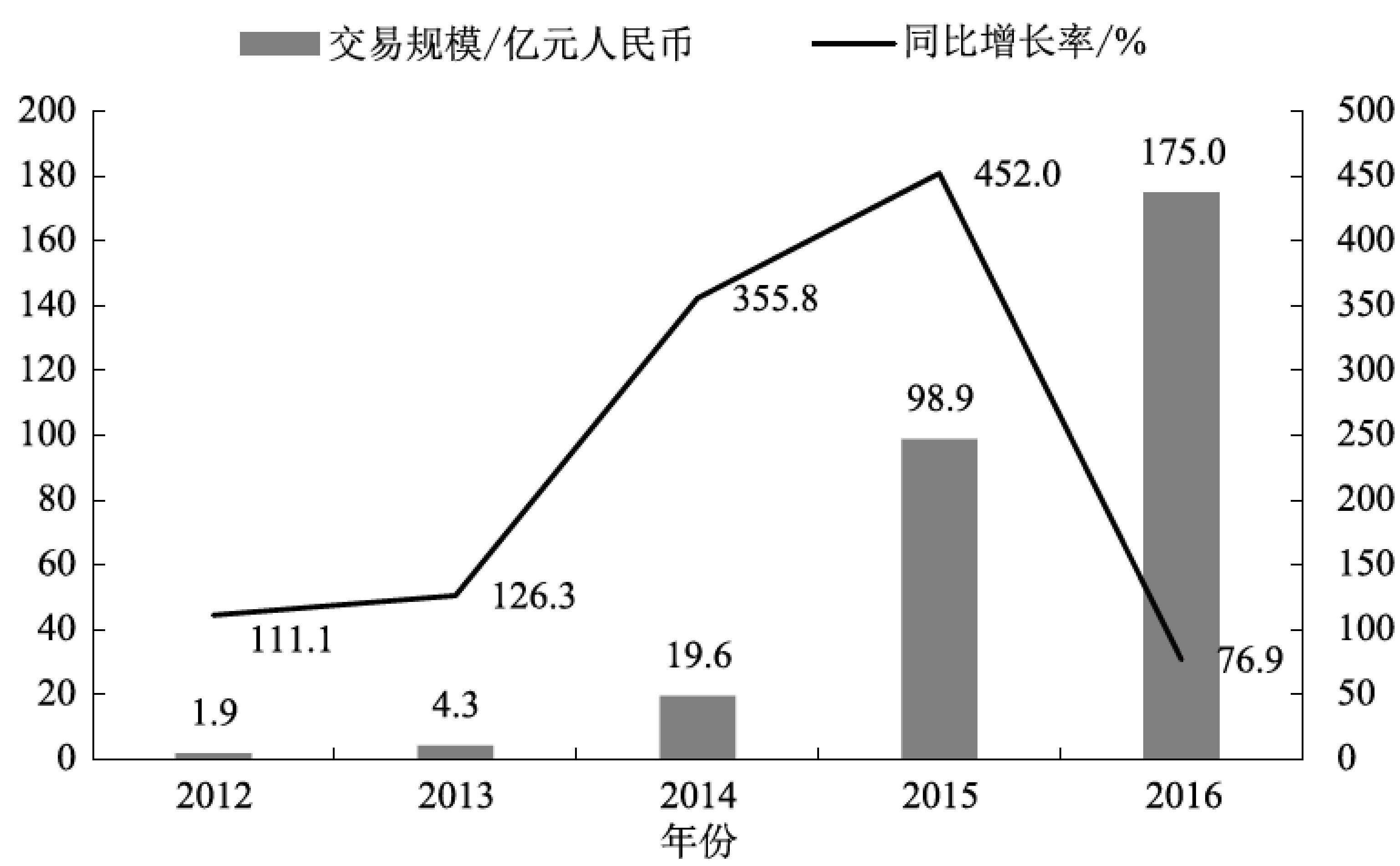


图 14.1　2012—2016 年中国众筹市场交易规模

资料来源：搜狐财经。

① 资料来源：宇博智业市场研究中心，《2016—2021 年众筹行业市场竞争力调查及投资前景预测报告》。

2）P2P 网贷

近年来，P2P 网贷领域经历了一个从混乱到有序，从高潮到低谷再回归理性的过程。2005 年，P2P 网贷平台在英国创立并迅速传入我国。2010 年前后，随着利率市场化、银行脱媒及民间借款的火爆，P2P 网贷在我国迅速发展，由于监管和法规的不健全，大量伪劣产品充斥市场，造成了我国金融秩序的混乱，P2P 网贷进入低谷。之后，随着法律法规逐步健全，行业监管逐步到位，我国的 P2P 网贷行业发展迎来了良性竞争、健康发展的阶段。据网贷之家数据显示，2016 年全年网贷行业成交量为 20 638.72 亿元，同比增长了 110%。其中，北京、上海、广东和浙江四地的网贷成交量占到了全国的 88.65%，显示出较强的“马太效应”。如表 14.1 所示，根据网贷平台发展指数排名，陆金所、宜人贷、点融网、人人贷和拍拍贷是目前影响力较大的 P2P 网贷平台。

表 14.1　2016 年 12 月网贷平台发展指数评级

排名	平台名	发展指数
1	陆金所	70.43
2	宜人贷	69.88
3	点融网	69.13
4	人人贷	68.66
5	拍拍贷	67.66
6	微贷网	65.86
7	搜易贷	63.96
8	爱钱进	63.7
9	开鑫贷	62.98
10	投哪网	62.92

资料来源：网贷之家、盈灿咨询。

注：发展指数由成交(8%)、人气(13%)、技术(5%)、杠杆(13%)、流动性(5%)、分散度(16%)、透明度(16%)、品牌(13%)和合规性(11%)九大因素构成。

3）第三方支付

作为一种新型的支付模式，国内第三方支付平台主要是以互联网为媒介的互联网支付和以手机为媒介的移动支付。支付工具则以手机扫描二维码、NFC(近距离无线通信技术)近场支付为主。自 2010 年以来，我国第三方支付市场的交易规模保持着年均 50%以上的增速，已然成为全球的领跑者。2016 年，我国第三方支付的交易规模已经达到了 57.9 万亿元，同比增长了 85.6%，其中移动支付交易规模为 38.6 万亿元，约为美国的 50 倍。[①] 如图 14.2 所示，按市场份额进行划分，2016 年我国第三方移动支付市场形成了三大梯队。具体来看，支付宝和财付通分别占据了 52.3%和 33.7%的市场份额，组成了第一梯队。拉卡拉、易宝、联动优势、连连支付、平安付、百度钱包、京东支付和快钱[②] 8 家有名的

① 数据摘自必达咨询所著《2016 中国第三方移动支付市场研究报告》。

② 拉卡拉、易宝、联动优势、连连支付、平安付、百度钱包、京东支付和快钱的市场份额分别为 2.7%、2.2%、2.1%、1.4%、1.2%、1.1%、1.0%、0.9%。

支付企业共同占据了13%的市场份额,形成第二梯队。其余的支付企业则组成第三梯队。

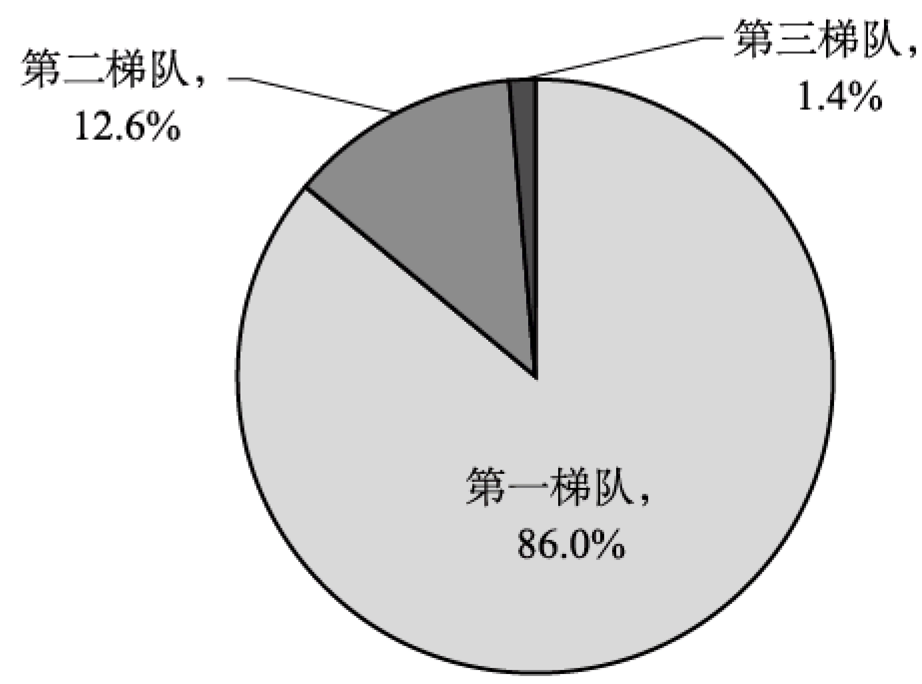


图 14.2　2016 年中国第三方移动支付市场份额

资料来源:必达咨询《2016 中国第三方移动支付市场研究报告》。

4) 数字货币

近年来,数字货币获得了极大的发展空间。CoinMarketCap 的数据显示,截至2017年6月,虚拟货币的种类已达885种之多,其中交易量活跃的有430余种,总市值超过1 000亿美元。在众多的数字货币中,最具代表性的当属比特币。此外,基于比特币设计的各类竞争币(在比特币机制上有所改进和变化的虚拟货币),如以太坊(ETH)、瑞波币(Ripple)和莱特币(LTC)等也正迅猛发展起来,迅速占领了一定的市场份额。截至2017年6月,比特币市值约占整个市场的38%,而以太坊、瑞波币和莱特币三者共同占到了约43%的市场份额。

随着数字货币进一步的规范化,未来数字货币必将获得进一步的发展。正如周小川所说:"数字货币的发展、区块链等新技术会在未来产生一些当前人们难以想象或者预测的影响。"

5) 互联网金融门户

随着互联网金融行业的发展,互联网金融门户网站如雨后春笋般涌现出来。根据商业模式划分,目前互联网金融门户网站可分为两类:第一类是垂直财经门户网站,可以凭借其专业化的信息提供,吸引诸多高端深度财经信息用户,如网贷界、东方财富、和讯网、金融界等;第二类是综合门户财经网站,可以凭借门户网站知名度和客户忠诚度,进入互联网金融信息服务行业,以开设财经频道的方式,细分其网站用户,如新浪财经、搜狐财经、腾讯财经等。现阶段,互联网金融门户可分为P2P网贷类门户、信贷类门户、保险类门户、理财类门户和综合类门户五大类,且无论是在内容还是形式上,都更加精细全面,并且同金融大数据的联系越发紧密。

专栏 14.1　P2P 平台如何催缴还款?基于行为理论的机制设计

行为经济学理论中有一个内疚厌恶效应(guilt aversion),是指在工作场景或者社会活动中人们会努力达到他人的期望,以降低自身内疚感。来自杜宁华、李玲芳等(2019)在一个自然实验中应用了内疚厌恶理论,探索如何提高P2P贷款平台的还款率。他们收集了一个贷款平台在10个月里贷款申请人的还款数据。这些贷款人通常被要求按月分期支

付还款。一旦出现逾期不还，平台会自动向贷款人发送催缴短信。在这个自然实验里，逾期的贷款人被随机分为三个组，分别收到三种不同内容的催缴短信。

三种催缴还款的短信内容如下：

1. 中性：*** 你好，由[平台名称]向你借的借款已到还款日期，请登录平台官网充值还款。

2. 正向期待：*** 你好，由[平台名称]向你借的借款已到还款日期，请登录平台官网充值还款。相信你一定会履行承诺，不让帮助了你的[平台名称]失望！

3. 后果威慑：*** 你好，由[平台名称]向你借的借款已到还款日期，请登录平台官网充值还款。再次提醒你按时还款，否则[平台名称]会向你采取行动！

实验结果显示，第二种短信，即表达了“正向期待”的短信产生了最好的催缴效果，该组的还款率最高；而“后果威慑”型短信并没有产生显著的提高还款率的效果。该研究表明正向期待短信启动了还款人的内疚感，增加了还款概率。利用行为理论进行机制设计在互联网金融领域具有广泛的应用空间。

资料来源：Ninghua Du, Lingfang (Ivy) Li, Xianghua Lu, and Tian Lu：Pro-Social Compliance in P2P Lending: A Field Experiment[J], Management Science(2019, forthcoming).

14.2 P2P 网贷中的行为金融

P2P 网贷借助互联网平台将借、贷双方结合起来。由于借、贷双方在网贷中所处的地位不同，接受到的信息量有限，在信息不对称和个人心理差异的影响下，P2P 网贷领域表现出一系列有趣的市场异象，主要包括羊群行为、面孔效应、社会关系、个人特征及信息不对称五方面。

1. 羊群行为

羊群行为又称从众行为、群体行为，指的是同大多数人一样思考、感觉、行动的人类社会现象。大量的研究表明，羊群行为普遍存在于我国的 P2P 网贷市场中。Zhang 等(2012)研究发现，在 P2P 网贷中，存在着理性与非理性这两类羊群行为。非理性的羊群行为指的是贷款者被动地做出模仿其他贷款者的行为，理性的羊群行为则指贷款者主动观察其他贷款者的决策行为，进而推测借款者的信用，并基于公开可观测的借款者特征来作出最优的行为选择。理性的羊群行为相比于非理性的羊群行为，能够更优地预测贷款的表现。基于网站数据的实证研究发现，理性的羊群行为在网贷市场上普遍存在。

网贷市场上的羊群行为显著地影响到了投资者的投资行为。Ceyhan 等(2011)研究发现，投资者在投标过程中存在着羊群行为，大多数投标数目的峰值产生在一个相近的时间点上。Herzenstein 等(2011)则发现，羊群行为强烈的项目，往往有着更佳的还款表现。此外，投资者的羊群效应也往往对借款订单的成功率产生较大的影响。

合理对待网贷市场上普遍存在的理性羊群行为，借款者应当通过项目内容、形式的设计，吸引更多投资者的目光，争取更多的投资人数；而投资者则应当关注有更多投标人数的项目，以保证贷款能够更及时地得以收回。

2. 面孔效应

P2P 网贷中，往往要求借款者提供自己的照片。因此，外貌特征往往影响到信贷的决策行为。

外貌对信贷的影响，往往是因为其影响了贷款人对借款者的主观信任程度。杨晓兰(2014)基于大学生的实验室研究发现，外貌上更具信任感和更开心的人，往往能够获得更好的贷款表现。另有学者在 MkurK 上招募任务执行人，对美国 P2P 平台 Prosper 网站上借款申请人的外貌进行量化，发现外貌可信任度高的借款申请人，得到贷款的利率低于平均水平。① 在网贷中，外貌往往能促成投资者对借款者形成第一印象。投资者从外貌中主观地了解到借款者的可信任水平及潜在的还款意愿、还款能力，进而凭借这些主观判断作出决策，形成面孔效应。

3. 社会关系

在 P2P 网贷中，存在着错综复杂的社会关系。社会关系往往影响到借贷双方的信任程度、借款意愿水平与信息不对称程度，进而影响贷款的实际过程。

朋友作为一种较为亲密的社会关系，改变了贷款过程中信息获取的程度与获取的难易程度，进而影响到整个贷款发生过程。基于 Liu 等(2015)研究发现，友谊在 P2P 网站中起到了重要的作用，借款人的朋友是借款人筹资的重要渠道之一，即存在渠道效应(pipe effect)。现实中的密友相较于一般朋友，现实中的一般朋友相较于网络朋友，对借款筹资发挥着更加显著的作用。同时，朋友间的借款会降低第三方潜在投资者的投资意愿，从而产生棱镜效应(prism effect)。此外，投资人更愿意跟随朋友，特别是密友进行投资，产生明显的羊群效应(herding effect)。

基于社区贷款的研究发现，P2P 网贷市场中存在着多层级声誉效应(multilevel reputation effect)。首先，进行高质量结构信息提供的社区，往往代表着更高的声誉水平，有助于贷款利率的降低。其次，进行高质量行为信号提供的社区，被认为有着更高的声誉水平，能保证贷款获得更低的利率。最后，拥有低质量个人信号的借款人，更能够借助社区的行为信号提高声誉，并从中受益。②

基于对群组关系与关系型借贷的研究发现，网络借贷关系中存在着套牢问题(hold up problem)。贷款人一旦掌握了借款人的私人信息，会垄断这部分的信息优势，并不会由于借款人违约风险的降低而降低借贷的利率。总的来说，套牢问题的存在使得私人群组的加入与关系型借贷并不能给予借款人更低的利率优势。③

4. 个人特征

借款人的个人特征如智力、口才等，能够反映出一定的可信任程度，代表了其潜在的

① DUARTE J, SIEGEL S, YOUNG L. *Trust and credit: the role of appearance in peer-to-peer lending*[J]. *Review of financial studies*, 2012, 25(8): 2455-2484.

② COLLIER B, HAMPSHIRE R C, *Sending mixed signals: multilevel reputation effects in peer-to-peer lending markets*. Proceedings of the 2010 ACM conference on Computer supported cooperative work, 2010.

③ EVERETT C R. *Group membership, relationship banking and loan default risk: the case of online social lending*. Banking and Finance Review, 2015.

还款能力，进而影响到 P2P 网贷市场。

语言中蕴含着借款人丰富的个人信息，直观地反映出一个人的能力与水平，进而影响其借款表现。一般而言，借款人展示的身份特征越多，信息披露越多，越能够成功获得借款。此外，就具体的身份特征而言，诚实可靠与成功的身份特征能够显著地增加筹资比例并获得利率上的优势，而明显的宗教信仰特征会降低获得融资的概率。诚信可靠与道德的特征往往带来较好的还款表现，而提及经济困难的借款人则存在违约还款风险。[②] 此外，借款申请的语言长度反映了借款人的信用与申请的努力程度，投资者能够根据语言长度很好地识别借款人的信用风险。而语言内容具有明显的迷惑性，投资者对语言内容的认识存在着一定的偏差。[①]

学历也是影响借款人借贷行为的重要特征。基于廖理等(2015)的研究发现，高学历的借款人具有较高的如约还清率与较低的信用风险水平，但高学历本身并未能获得更好的借款条件。这说明教育具有规范受教育人行为、降低违约概率的作用，但学历并不是借贷市场中考虑的主要因素。

借款人的学习行为也是影响借贷关系的一个重要因素。投资者过往的投资经验能够显著影响其当前的投资行为。一方面，存在学习行为的投资者能够主动调整自己的投资时机。一般而言，随着投资经验的累积，投资者更会等到借款标的的晚期再进行投资。另一方面，存在学习行为的投资者也能够主动调整自己的投资选择。一般而言，随着投资经验的累积，投资者更会选择信用等级高的标的进行投资。[②]

借款人的身份特征及现实的社会关系会大大影响其借贷行为。借款人拥有的朋友类型，朋友是否具备贷款记录以及他们参与借款人借款申请的投标与否，都会显著地影响借款成功率、借款利率以及违约的概率。一般而言，基于“名誉成本”(social stigma cost)的影响，有违约倾向的借款人会尽量避免在 P2P 平台上建立朋友关系。而借款人一旦在平台上拥有了朋友，将保持较高的信誉水平并相应地降低借款利率和违约概率。[③]

5. 信息不对称

在 P2P 网贷中，借贷双方间存在着信息不对称的情形，主要体现在信息甄别与信息披露两方面。

贷款者为了缓解借款过程中的信息不对称度，保证贷款能够安全、及时地收回，会对借款者的信用能力进行甄别。Iyer 等(2009)发现，贷款者有一定的推断同一信用等级内借款者信用差异的能力，并给予高信用评分的借款者以更低利率的优惠。借款者信息中的一些标准银行变量(如现有负债、负债额、收入负债比、过去 6 个月的信用问询次数等)，对贷款者的信用甄别能力影响较大。

---

① 廖理，吉霖，张伟强．语言可信吗？借贷市场上语言的作用——来自 P2P 平台的证据[J]．清华大学学报，2015.

② 主要参考：王正位，向佳，廖理，等．互联网金融环境下投资者学习行为的经济学分析[J]．数量经济技术经济研究，2016.

③ LIN M，PRABHALA N，VISWANATHAN S．*Judging borrowers by the company they keep：friendship networks and information asymmetry in online peer-to-peer lending*[J]．*Management science*，2013，59(1)：17-35.

借款者在利用P2P平台筹措资金过程中，也会被要求具体的信息披露，以降低信息不对称性。Michels(2012)研究发现，借款人披露不可验证的信息，能够显著地降低借款利率，增加竞标的人数。信用等级越低的借款人，越会受到不可验证信息披露的影响。此外，信息披露程度越高，往往意味着更低的信用违约率。

14.3 众筹中的行为金融

众筹，是以团购和预购的形式，向网络投资者募集项目资金的一种融资模式。由于项目的筹资者与投资者在信息掌握程度、个人特征及社会关系等方面的差异，众筹过程中同样存在着一系列的市场异象，主要体现在信息不对称、个人特征及社会互动三方面。

1. 信息不对称

在众筹过程中，筹资者获得足额项目资金的关键是对项目的陈述及引起投资者的关注，而投资者进行投资的最主要依据是以项目为基础的各类信息。众筹过程中主客观上的各类因素，影响了众筹中投融双方的信息掌握程度，产生了各类极富差异的众筹行为。

众筹参与者有着一定的隐私偏好，不愿个人信息被公开。Burtch 等(2015)的研究发现，减少众筹过程中的信息控制能够很好地提高众筹的参与意愿，但会使众筹金额的平均值减小。在众筹中提示有关投资者的隐私问题，会通过启动效应[①](priming effects)引起投资者对隐私的担忧。因此，改变众筹网站的设计机制，增加对相关人员信息隐私的保护，将对众筹过程产生显著的影响。

众筹项目的质量参差不齐，而投资者往往无法获得足够多的准确信息。将众筹分成商品众筹模式与股权众筹模式两类进行考虑发现，在信息不对称及产品质量不确定的情形下，商品众筹模式的融资机会会减少，而股权众筹模式下的融资机会则会上升。可见信息不对称程度大大影响了投资者对产品质量的信心，并将关注度从商品融资转移到了股权融资中。[②]

信号传递理论能够对众筹行为很好地作出解释。众筹中的中介平台通过展示创业项目的信息，显示项目的质量，在一定程度上缓解了投融双方的信息不对称性，进而影响到项目融资绩效。郑海超等(2015)研究发现，众筹过程中主动发布项目更新信息，在项目介绍中突出团队实力、团队规模以及同投资人(特别是领头人)积极沟通等行为，都能够释放出正面的信号，提高相应的融资绩效。此外，线下路演也是传送融资信号的一大有效途径。

2. 个人特征

众筹行为深受投融双方个人特征的影响，这种影响表现在投融双方主客观的各方面。

地理因素能够对项目融资产生很深的影响。比如投资者的投资时间选择。项目发起

---

① 启动效应是指由于之前受某一刺激的影响而使其以后对同一刺激的影响变得敏感的心理现象。

② BELLEFLAMME P, LAMBERT T, SCHWIENBACHER A. *Crowdfunding: tapping the right crowd* [J]. *Journal of business venturing*, 2014, 29(5): 585-609.

者的当地人更愿意选择在项目融资的前期进行投资，而外地人则更偏好于在项目融资的后期进行投资。这是由于本地的投资者同项目发起者有着更紧密的社交关系，对项目的信息掌握程度更加充分，因此可以更早地做出融资的行为。①

众筹行为同时表现出了国别的差异，这其中各国的文化差异扮演了重要的角色。Zheng(2014)对比了中美的众筹平台，发现中国企业的社会资本对众筹绩效的影响更加强烈。这种差异同中美间的文化环境差异有关。中国社会强调“关系文化”，有着显著的集体主义观念和“熟人”理念，这让中国的众筹行为更为重视义务和责任感，使得社会资本在众筹绩效中发挥了极大的作用。

众筹项目本身存在着一系列个性特征，反映了融资方的个人特点，进而影响到投资者的投资决策。黄健青等(2015)对国内众筹平台“追梦网”进行研究，基于顾客价值理论分析发现，项目回报、发起人历史发起项目数量、发起人网站积分、图片数量、项目的分享数量和评论数量的提升，都能够增加项目筹资的成功率。而项目的目标筹资额则对项目筹资成功率有着负面的影响。投资者往往是从一个顾客的角度，审视众筹项目这一产品的质量好坏，而众筹项目及融资方的系列个体特征，往往在投资者的投资决策中起到关键作用。

专栏 14.2 众筹的中美差异

首先，项目类型有差异。美国的众筹项目集中体现在创意项目，以科技产品为主导。而我国多为各类企业融资或者是实体行业投资项目，创意类项目较少。

其次，投融资环境不同。相比于美国复杂而多层次的融资市场，我国金融产品相对匮乏，特别缺乏针对中小企业及创业型企业的金融产品。正是基于上述原因，股权众筹模式在中国被快速复制。

最后，监管环境不同。美国的众筹环境较为成熟，已经出台了各类法案使众筹合法化。而我国众筹的法律法规尚需加以完善，监管尚需进一步加强。

资料来源：http://www.renrentou.com/.

3. 社会互动

社会互动影响了众筹投融资双方间信息的相互了解程度及可信任度，不同程度的社会互动导致投融资双方作出一系列不同的价值取向和收益判断，产生多样的众筹行为。众筹项目中的社会互动主要包括投融资双方的社交关系以及项目本身的自反馈效应两个方面。

良好的社交关系往往使得投融资双方间有着频繁、友好且持续的社会互动，能够令投资方掌握更丰富的众筹项目信息，减少了其信息识别的难度，增加了项目融资的成功率。Agrawal 等(2015)对众筹平台 Sellaband 上的所有投资记录数据进行研究发现，众筹项目投融资双方间的社交关系越紧密(表现为彼此更加了解，地理距离更为相近)，投融资双方的社会互动过程便越顺畅，投资者对项目的识别过程也更为迅速，会更早地做

---

① AGRAWAL A, CATALINI C, GOLDFARB A. *Crowdfunding: geography, social networks, and the timing of investment decisions*[J]. *Journal of economics & management strategy*, 2015, 24(2): 253-274.

出投资行为。

而就众筹平台本身而言，良好的平台机制结构设计，能够改善平台同投资者间的社会互动状况，进而增加成功融资的概率。众筹平台的诸多机制均对融资项目成功有着正面影响。首先，“筹资达到目标才能拨付资金”的阈值机制降低了项目发起人欺诈的可能性；其次，尽职的调查机制和质量信号能够剔除不切实际的冒险项目；最后，众筹社区的反馈机制能够降低投融资双方的信息不对称，进而提高参与双方的积极性。众筹机制的上述机制，使得投融资双方间的社会互动更加正面而紧密：一方面，使该项目能够得到更合理的评估，降低欺诈和违约风险，并利用“负反馈效应”淘汰劣质项目；另一方面，让优质的项目获得更多投资者的关注与支持，从而加强了众筹的“正反馈效应”。基于众筹平台“点名时间”的实证研究发现，筹资金额较小，具有优质信号的项目有着更良好的社会互动关系，更容易成功进行融资。[①]

延伸阅读 14.1　众筹中的群组制度

Hildebrand 等(2017)研究了众筹的群组制度。作者认为，众筹的群组组长往往能够获得推荐回报的机制，会使得部分组长在众筹标的的推荐上基于自身利益最大化而非标的风险出发，从而使得其他投资者难以识别组长推荐标的的真实风险，进而加剧众筹市场的信息不对称程度。利用 Prosper 众筹平台(Prosper.com)上的数据，作者研究了不同种类的群组，分析不同群组组长推荐标的的风险差异，进而研究不同群组的激励模式。首先，作者将 Prosper 平台中的群组分为有回报群组和无回报群组，并将研究集中于有回报群组在回报机制取消前后的表现，以验证订单获得组长投标在不同情况下的风险揭露能力。其次，文章对订单成功率、利率和违约率分别采用 logit、tobit 和 cox 模型进行研究，并引入双重差分法分析不同群组的订单表现。最终实证研究的结果表明，无回报群组中组长投标能够增加成功率，降低利率和违约率；有回报群组中组长投标能够增加成功率并降低利率，但只有在取消回报或组长拥有足够份额下才能降低违约率。

资料来源：HILDEBRAND T，PURI M，ROCHOLL J. *Adverse incentives in crowdfunding*[J]. *Management science*，2017，63(3)：587-608.

14.4　亲社会网贷中的行为金融

亲社会网贷是按照社会希望的方式进行网络贷款的一种互联网金融筹资模式。相比于 P2P 网贷，亲社会网贷更具有公益的性质，贷款的对象往往面向社会弱势群体以及发展中国家和地区，贷款的利率较低，还款压力较少，同“普惠金融”有共通的特点；相比于众筹，亲社会网贷并不以投资为最根本的目的，同时还关注融资方的现实需求。亲社会网贷本身的网贷属性以及所独有的公益性质，使得其既有同 P2P 网贷和众筹相类似的行为金

① 黄玲，周勤．创意众筹的异质性融资激励与自反馈机制设计研究——以“点名时间”为例[J]. 中国工业经济，2014(7)：135-147.

融特征，也有着独属于自己的不同特点。

有关亲社会网贷的研究大多借助 Kiva 平台进行。Kiva 是一家非营利私对私小额贷款机构，以向发展中国家的创业者提供小额贷款，进而消除贫困为目标。Kiva 的资金来源于有空闲资金的个人，这些投资者并不注重经济回报，而更在意精神回馈，因而能够以从事慈善事业的心态去提供资金帮助。Kiva 通过网络获取资金，并以免息或极低的利息给予小额贷款机构资金，并借由这些机构贷款给穷人。

基于 Kiva 的研究发现，在亲社会网贷中，社会认同、个人特征与羊群行为是最明显的行为金融特征。

1. 贷款动机分析

亲社会网贷贷款人（资金提供方）的贷款动机主要基于贷款人贷款陈述语言进行研究分析。在广泛的研究中，学者往往将贷款人关于贷款原因的陈述语句进行量化：先随机选择小部分贷款陈述语句进行语意概括，提炼贷款人提供贷款的动机，并形成一定的分类标准，再分别借助人工判读和机器学习的方法，将海量的贷款人陈述语句全部进行标准化分类，最终得到量化的文本分类数据，进行实证研究。基于上述方法，Liu 等（2012）研究发现，不同的贷款动机能够显著地影响贷款人的贷款频率与贷款额度。出于利他主义和外部原因的贷款人往往提供贷款的频率较低，而看好平台发展和出于宗教信仰目的的贷款人有着更高的投资频率和更大的贷款额度。此外，加入团队的贷款人往往有着共同的价值追求和稳定明确的贷款目标，能够比不加入团队的贷款人提供更大的贷款数额。

专栏 14.3　Kiva 贷款的文本分类

在 Liu 等（2012）所写的“*I loan because*…”：*understanding motivations for pro-social lending* 一文中，作者先从众多的贷款人贷款陈述语句中随机抽取 200 条进行语意分类，共分成了利他主义（general altruism）、同情（empathy）、互惠（reciprocity）等十类，之后雇用了学生按照相同的标准将全部的 5 250 条陈述语句依据语意分成十类。为了保证分类结果准确，作者又采用了机器学习的方法，采用朴素贝叶斯、最大熵和支持向量机等不同的计算机算法进行语义分类。

最后的语义分类结果显示，出于利他主义、社会责任和社会规范（Social responsibility and social norms）动机提供贷款的投资者人数最多，而出于同情、互惠动机的贷款人最少。

资料来源：LIU Y，CHEN R，CHEN Y，et al. “*I loan because*…”：*understanding motivations for pro-social lending* [C]. Acm international conference on web search & data mining，2012：503-512.

2. 社会认同

社会认同对个人有着激励作用，往往能够改变个人的办事风格，提高个人的办事效率。反映在亲社会网贷领域，社会认同能够让一群人为了共同的目标去投融资，提高网贷的规模与成功率。亲社会网贷中的社会认同可分为群体身份与团队竞争两部分。

一方面，有群体身份的贷款人在一个共同的群体中会进行相互学习，进而增加对群体身份和群体目标的认同，提供更多的贷款数额。另一方面，加入团队的贷款人相比于没有加入团队的贷款人，能够提供更多的贷款额度。这是因为，有群体身份以及加入团队的贷

款人，接触到了明确的贷款目标，并为了目标的实现在内部进行信息的协调，从而在社会认同的激励下，会显著地增加贷款的数额。

3. 个人特征

投融资双方的个人特征会显著地影响到亲社会网贷行为，文化差异和地理距离是表现最为突出的两类特征。

基于文化和地理因素，亲社会网贷行为上存在三大效应：首先是文化效应，贷款人同借款人间的文化差异越大，彼此之间的信任便会越少，贷款人会降低贷款意愿，从而使贷款交易减少。文化差异主要是通过影响借贷双方间的信任程度发生作用的，而第三方机构信托机制能够降低文化效应的影响。其次是地理效应，贷款人同借款人间的地理距离越大，贷款人往往对借款人越不熟识，这也会降低贷款人的贷款意愿，从而使贷款交易减少。最后是替代效应，地理距离和文化差异是可以相互替代的，导致存在着缓和效应，即随着地理距离的变大，文化因素对贷款行为的影响会下降，反之亦然。这是因为，随着距离的扩大，贷款人更不容易了解借款人所在国家的文化，从而导致文化意识的缺失（文化盲目性），而在没有文化意识的条件下，一般认为个人会倾向于认为彼此间的文化是相似的。

除了上述两大特征外，借款人所在国是否发生自然灾害，以及贷款人所在国的人口多样性和外国出生居民数等借贷双方外部特征，也会显著地影响到贷款的数额。[①]

4. 羊群行为

亲社会网贷中的贷款人同样表现出明显的羊群行为。吴佳哲（2015）在 Kiva 平台上随机选取了 1 000 笔原始借款数据进行分析发现，亲社会网贷中的贷款人行为存在显著的羊群效应，且偏好“已筹集百分比”较高的借款项目。此外，“借款期限”“借款金额”和“还款方式”等因素也会对贷款人的行为产生显著的影响。

从信息不对称角度来看，借款人的信息缺乏真实性保障以及贷款人融资缺乏专业性，导致了借贷双方间存在严重的信息不对称性，进而产生了严重的羊群行为。一方面，信息不对称使得贷款人不了解亲社会贷款平台的项目信息，为了规避风险，贷款人倾向于投资已获取较大融资金额以及融资速度较快的项目，从而表现出羊群行为。另一方面，贷款人往往缺乏专业的投资经验，为了保证获取理想的投资回报，贷款人偏好采取模仿策略提供贷款，从而产生羊群效应。

基于亲社会网贷中普遍存在羊群行为这一现实，网贷平台应当从网页设计、信息挖掘等方面提高平台质量，确保提供的信息更为充足全面，便于贷款人了解，从而使贷款人在信息更为充分的情况下更乐意去提供更多的资金，进而增加贷款项目的融资成功率。

延伸阅读 14.2　社交网络中的羊群行为

在互联网金融中，社交网络正发挥着日益重要的作用，深刻影响着投资者的投资活

① BURTCH G，GHOSE A，WATTAL S. *Cultural differences and geography as determinants of online prosocial lending*[J]. *Management information systems quarterly*，2014，38(3)：773-794.

动。Cao 等(2014)分析了在存在社交网络的情形下,基金经理基于信息获取的羊群行为。

首先,基于 Kyle(1985)的经典模型,文章建立了基于信息获取的羊群行为模型,研究发现当基金经理的信号为互补性信号时,基金经理人之间便会发生羊群行为,但是竞争的存在导致交易同一只股票的基金经理不能超过 3 个。

其次,在存在社交网络和信息共享机制的情况下,羊群行为能够在任意数量的基金经理人之间发生,信息的共享扩大了羊群行为的参数集,而带噪声的信息共享又会进一步扩大羊群行为的参数集。

最后,在动态交易情形下,羊群行为发生的参数区间缩小,而在连续时间交易情形下,激烈的竞争使得经理单独投资的收益总是大于羊群行为的收益。一般而言,在不存在社交网络时,经理人们不会发生羊群行为。

该文存在的不足之处是,出于模型简化的考虑,仅仅考虑了交易单只股票的情形,忽略了经理对多只股票拥有多个信号的情形。此外,该文仅仅考虑了共享同质信息的情形,而忽略了对不同个体分享异质信息的情形。

资料来源:CAO H H,YE,D. *Social network,herding and competition*[C]. *China international conference in finance*.

14.5 其他互联网金融模式中的行为金融

1. 第三方支付

第三方支付离不开网络的技术支持,而网络上层出不穷的安全事件一直影响威胁着第三方软件的发展,也使得第三方支付软件的使用者面临较高的使用成本。

以余额宝为例的研究发现,用户对网络安全风险的感知程度能够影响互联网金融资产的定价。曾建光(2015)采用余额宝被盗相关主题的百度大数据来衡量互联网金融投资者的网络安全感知程度,同时设置了多个有关于余额宝收益率的变量,进而构建了实证模型进行研究发现,投资者的网络安全风险感知程度越高,风险补偿越多。同时,从不同的终端考虑,相比于 PC 端(个人计算机),移动互联网端的投资者往往要求更多的风险补偿。此外,针对余额宝的研究进一步发现,阿里巴巴在美上市之前、基金经理人采用激进投资策略时、市场资金面宽松情境下、投资者为使用非苹果智能手机的投资者以及投资者为 PC 端女性投资者时,投资者的风险感知程度同风险补偿之间的正相关关系得到加强。

第三方支付正处于飞速发展的过程中,随着其规模越来越庞大,越来越突出的安全问题增加了第三方支付的使用成本,影响了其本身的资产定价。如何制定及时有效的政策以及提升监管效率,将是第三方支付未来发展所面临的一大突出难题。

2. 数字货币

近年来,数字货币逐步兴起,价格水涨船高,随着各类数字货币的实际价值逐步被现实价格所超过,数字货币领域的泡沫也逐渐显现出来。

一开始,数字货币数字化的产生方式及对抗现有货币体系的设计初衷吸引了一小部

分人进行投资并传播。随着越来越多的人开始了解和信任数字货币，更多的投资者将其作为了一种可信赖的投资工具。随着购买投资数字货币的人越来越多，在专家乐观预测及新闻媒体的大肆炒作下，数字货币市场的羊群效应便逐渐显现。数字货币的需求量越来越高，推动了一批又一批的投资者进入数字货币领域，并使得数字货币的价格开始超越价值。在狂热的投资热情下，看多的势力完全盖过了看空的群体，从而进一步推动了数字货币市场的升温，市场泡沫逐步形成。

近期以来，数字货币依旧有着升温的趋势，泡沫还在持续。邓雪莉(2015)认为，数字货币之所以产生巨大的泡沫，是“个体投资者行为模式中的心理偏差演化成了系统性的群体偏差，最后导致了异常的市场表现，然后又通过正反馈机制影响个体的信念，强化了行为偏差，最终引发非理性的市场狂热”。对数字货币市场进行监管与规范，是下一阶段数字货币发展的重中之重。

专栏 14.4　互联网金融中的“情绪黏性”与心理账户

互联网金融市场中，常常可以依据投资者的“情绪黏性”与心理账户制定可行的投资策略。

投资者情绪，主要反映了投资者在投资行为中的主观意愿，对股市预测能产生重要的作用。一般而言，投资者情绪指标可分为直接指标、间接指标和情绪代理变量三类。直接指标往往通过调查问卷等方式来获取；间接指标有换手率、市盈率、市净率等反映投资者心理变化的变量；情绪代理变量包括天气、气压、云层覆盖率等外部环境因素。金辉(2016)基于投资者情绪的研究发现，互联网金融板块上存在着投资者的“情绪黏性”，即投资者因为倾向于习惯性操作、过度自信以及从众行为，不能及时地根据客观事实转变主观态度，在改变投资策略上具有一定的滞后性，但这种“情绪黏性”持续时间较短。

不同投资者的预期收益存在差异，因此他们在投资过程中会设置多个心理账户，并在决策时做出一些违背经济运算法则的非理性行为。胡国生(2015)基于储蓄(S)和投资(I)构建了货币盈余(P)，并引入价值函数$Y(*)$和决策权重函数$\rho(*)$构造了心理账户函数。继而对心理账户进行分析发现，基于安全角度的考虑，目前投资者投资比例与金额有限，普通大众总在收益与安全之间进行权衡，存在一种尝试心理，并倾向于做出从众的行为。为消除投资者目前普遍存在的心理账户，政府需要参与监管，整个互联网金融市场也亟待完善。

资料来源：金辉．互联网金融股票投资中的投资者“情绪黏性”[J].中国集体经济，2016(21)：88-89.

胡国生．基于行为金融的互联网金融分析[J].武汉金融，2015(1)：58-59.

课程实验指引[①]

1. 实验背景

本实验将建立一个 P2P 网贷市场，每个参与人作为一名贷款人或借款人，在市场中进

① 本实验基于 trust game 设计，实验目的在于检验借款人和贷款人之间的信任关系。

行交易。在每轮实验开始前，贷款人都会被赋予一定数量的账面现金作为其拥有的初始资产。贷款人同借款人的资产会根据其在实验中的投资决策相应增减，并同最终获得的收益相挂钩。具体而言，实验将按照如下的方式进行。

2. 实验周期

本实验共10轮，也就是有10个交易周期。在10个交易周期中获得的实验收益，将同每个贷款人和借款人最终所获得的报酬相挂钩。

3. 交易资产

每个贷款人在每轮实验开始前，都有账面现金资产30点。在实验中，每个贷款人可以从账面现金资产中选择投资 y 点($0\leqslant y\leqslant 30$)给借款人，借款人自动获得 $3y$ 点的资产后可以选择返还 x 点($0\leqslant x\leqslant 3y$)给贷款人。在每轮结束后，贷款人拥有的现金资产为 $30-y+x$ 点，借款人拥有的现金资产为 $3y-x$ 点。

4. 交易顺序

在每轮实验中，每个参与人的贷款人或借款人身份保持不变，贷款人同借款人两两随机分成一组进行实验。实验开始后，每一轮贷款人都将同不同的借款人分为一组。具体的分组方式将由系统随机决定。

5. 交易机制

在实验开始前，每个借款人将输入自己的性别信息。

在实验的第一轮，贷款人在作出决策前，将看到自己配对的借款人的性别信息，并基于该信息选择投资多少点数给借款人。借款人基于贷款人投资后自动获得的3倍点数，再选择留存与返还的点数。

从第二轮开始至第十轮，在每一轮中，贷款人在作出决策前，都将看到自己配对的借款人的性别信息以及借款人上一轮留存与返还的点数信息(借款人上一轮的决策历史记录)，并基于上述信息选择投资多少点数给借款人。借款人再基于贷款人投资后自动获得的3倍点数，选择留存与返还的点数。

在每一轮中，贷款人匹配到的借款人都是不同的。每一轮结束后，贷款人和借款人都会知道本轮他们最终拥有的现金资产是多少点。

6. 实验报酬计算

实验报酬按照如下的方法计算。

在每一轮实验中，贷款人从30点账面现金资产中投资 y 点给借款人，并获得借款人返还的 x 点资产，最终每轮拥有的现金资产是 $30-y+x$ 点。借款人从在贷款人投资 y 点后自动获得 $3y$ 点资产，选择返还 x 点给贷款人后，最终每轮拥有的现金资产为 $3y-x$ 点。

下面举一个简单的例子来说明实验报酬的计算方法。

如果在某一轮中，贷款人从30点账面现金资产中投资20点给借款人，借款人自动获得60点现金资产后返还30点资产给贷款人。那么，最终贷款人将获得30点－20点＋30点＝40点的现金资产，借款人将获得60点－30点＝30点的现金资产。

最终，在10轮实验中，将随机抽取1轮，得到该轮交易结束后贷款人和借款人拥有的

现金资产，并按照一定比例，将实验中的点数化为人民币，计算得到每个贷款人和借款人在该轮实验中的收益，这也是每个贷款人和借款人在本实验中得到的最终报酬。

本章小结

（1）互联网金融是指传统金融机构与互联网企业利用互联网技术和信息通信技术实现资金融通、支付、投资和信息中介服务的新型金融业务模式。

（2）综合来看，目前互联网金融的发展模式有众筹、P2P 网贷、亲社会网贷、第三方支付、数字货币、大数据金融、信息化金融机构、互联网金融门户八大类。

（3）P2P 网贷领域，主要存在羊群行为、面孔效应、社会关系、个人特征及信息不对称五方面的行为金融特征。面孔效应指的是外貌会影响贷款人对借款人的主观信任程度及最终的贷款决策。渠道效应指的是借款人的朋友是借款人筹资最为重要的渠道之一。棱镜效应指的是朋友间的借款会降低第三方潜在投资者的投资意愿。

（4）众筹领域，主要存在信息不对称、个人特征及社会互动三方面的行为金融特征。启动效应指的是由于之前受某一刺激的影响而使其以后对同一刺激的影响变得敏感的心理现象。

（5）亲社会网贷领域，主要存在社会认同、个人特征与羊群行为三方面的行为金融特征。文化效应指的是贷款人同借款人间的文化差异越大，彼此之间的信任便会越少，贷款人会降低贷款意愿，从而使贷款交易减少。地理效应指的是贷款人同借款人间的地理距离越大，贷款人往往对借款人越不熟识，这也会降低贷款人的贷款意愿，从而使贷款交易减少。替代效应指的是地理距离和文化差异是可以相互替代的，导致存在着缓和效应，即随着地理距离的变大，文化因素对贷款行为的影响会下降，反之亦然。

（6）在第三方支付和数字货币领域，同样存在着各类行为金融特征。

思考题

1. 各类互联网金融模式有哪些共同的行为金融特征？

2. P2P 网贷中的羊群行为有哪些表现？请举例说明。

3. 众筹中的个人特征有哪些表现？请举例说明。

4. 试说明 P2P 网贷、众筹与亲社会网贷这三类互联网金融模式在行为金融特征上表现出的共同点与区别。

5. 如何看待数字货币存在的泡沫问题？如何衡量数字货币中存在的泡沫规模？

第 15 章

决策、不确定性和大脑：神经金融学

引导案例：大脑是怎样产生金融决策行为的

如果你发现刚刚卖出的股票大涨，或者发现没有买的股票收益却很高，你会感觉后悔吗？你会想要再把它买入吗？后悔是一种主观负面情绪，每个人的后悔体验不尽相同，因此现实中很难去定量衡量后悔。Frydman 和 Camerer(2016)却通过神经科学的方法得出了投资者后悔情绪与其股票交易行为之间的联系。

Strahilevitz，Barber 和 Odean(2011)发现了“回购效应”(repurchase effect)，相比于在股票卖出后价格下降时买回该股票，投资者更不可能在同一股票卖出后价格上涨时买回该股票。于是，Frydman 和 Camerer(2016)通过神经金融学实验检验了后悔情绪对个人投资者回购行为的影响，从后悔理论的角度解释了“回购效应”。在实验中，股票的价格被设置成正自相关的，这就意味着，当投资者卖出股票后发现股票价格上涨了，由于股票价格的正自相关性，股票价格在未来依然会涨，理性投资者应该买回该股票；而投资者卖出股票后发现股票价格下跌了，由于股票价格的正自相关性，股票价格在未来依然会跌，理性投资者不应该买回该股票。因此在这样的实验设计下，理性投资者应该拥有与“回购效应”相反的表现。然而实验结果表明，被试表现出比遵循理性决策情况下更强的“回购效应”。他们使用功能性磁共振成像(functional magnetic resonance imaging，fMRI)记录下被试发现他们刚卖出的股票价格上涨时的大脑活动的数据，在大脑有关报酬机制的脑区中发现了后悔信号。被试后悔信号越强，其“回购效应”越强。因此，他们通过神经科学的方法定量衡量了投资者后悔情绪，并发现后悔显著影响投资者后续的交易行为。

15.1 神经金融学简介

金融决策一直是金融学、心理学、神经科学等诸多学科领域的学者共同探索的主题。

新古典主义经济学将人的大脑看作一个黑箱，提出“理性经济人”假设，并在此基础上建立理性选择模型，但却严重忽视了大脑的活动以及心理对个体金融决策行为的影响。尽管主流金融学对诸多实际问题都作出了合理的解释，但是金融市场仍然存在诸多异象，对主流金融学理论提出了一定的挑战。20 世纪 80 年代，行为金融学应运而生，它通过借鉴大量心理学的研究成果来研究个体金融决策行为，指出人是有限理性的。而神经金融学作为一门新兴的交叉学科，它更为深入地研究大脑活动，通过借鉴心理学、神经科学以及金融学领域的诸多成果，试图揭示大脑在作出复杂金融决策时的细节，以及大脑活动和个体金融决策之间的关系，由此建立一种更加准确与完善的决策模型。神经金融学的研究结果表明，同一类型金融决策往往与大脑某个特定区域活动息息相关。这对于深入理解金融决策中人们完整的认知、思考及决策过程具有重要的意义。

神经金融学是一门将金融学和神经科学与心理学相结合而形成的新兴交叉学科，其在行为金融学的基础上，继续补充完善传统金融学理论，从神经学层面为人的决策行为提出佐证。神经金融学通过神经科学的工具来研究金融市场中的决策行为，从基础的生理层面挖掘对决策有重要影响的生理学变量，并借此对人们的金融决策行为，尤其是传统金融学无法解释的内容作出科学合理的解释。此外，神经金融学也为行为金融学的一些理论建立了初步的神经学基础。

神经金融学主要的研究内容是利用神经科学的设备，来研究与金融学密切相关的主题，如风险、情绪、博弈、价值、动机、过度自信、各类偏差和异象等。神经金融学希望通过理解大脑活动的内在机制，对金融行为作出更合理、更基础的解释。基于面临选择时的大脑活动，神经金融学正试图提供一个新的经济决策模型。神经金融学使用的工具，允许通过分析作出这些选择时的大脑活动情况理解人们的经济决策行为。通过观察大脑的内部活动，可以建立更加实际的决策模型，并能够解释比传统标准模型更广泛的个体经济行为。

神经金融学运用的神经科学设备主要有功能性磁共振成像、事件相关电位（event-related potential，ERP）、经颅直流电刺激（transcranial direct current stimulation，tDCS）、经颅磁刺激（transcranial magnetic stimulation，TMS）、脑磁图（MEG）等，此外也会结合心理测量、神经药理学、荷尔蒙测定等多种手段。神经科学方面的设备和手段的迅速发展，大大激发了神经金融学研究者探究大脑内在机制及其与人们金融决策行为之间的关系的兴趣。在神经金融学的发展中，技术的发展起到了重要作用，“新的工具定义了新的科学领域，并消除了旧有边界”[①]。表 15.1 简单介绍了几种常用的神经金融学设备的作用原理。

神经金融学近年来发展迅速，众多学者投身此领域，借助交叉学科强大的创新能力，已经取得了不少成果，陆续在国际尖端杂志诸如 *Science*，*Nature* 等发表。然而，人类行为有着令人困惑的复杂性和异质性。神经金融学仍然需要漫长的努力去寻找、探索金融决策行为时的大脑活动的内在机制和规律。

---

① CAMERER C F, LOEWENSTEIN G, PRELEC D. *Neuroeconomics: how neuroscience can inform economics* [J]. *Journal of economic literature*, 2005, 43(1): 9-64.

表 15.1 几种常用的神经金融学设备的作用原理

英文简称	fMRI	ERP	tDCS	TMS
全称	功能性磁共振成像	事件相关电位	经颅直流电刺激	经颅磁刺激
作用原理	神经细胞活化时会消耗氧气，而氧气要借由神经细胞附近的微血管以红血球中的血红素运送过来。故可通过血氧浓度依赖(BOLD)来侦测脑中的反应区域	通过平均叠加技术从头颅表面记录大脑诱发电位来反映认知过程中大脑的神经电生理改变	利用恒定、低强度直流电(1～2 mA)调节大脑皮层神经元活动，阳极刺激通常能增强刺激部位神经元的兴奋性，阴极刺激则降低，伪刺激多作为一种对照刺激	磁信号可以无衰减地透过颅骨而刺激到大脑神经，高频(>1 Hz)磁信号主要是兴奋的作用，低频(≤1 Hz)磁信号则是抑制的作用

15.2 不确定性相关的金融决策

不确定性下的决策行为是经济学领域重要的研究命题。现实中的金融行为经常涉及不确定条件下的决策问题，投资者要选择何时买卖哪只股票，但股票价格却不可预测，公司 CEO 要决定投资哪个项目，但项目的回报率却并不确定。

Knight(1921)区分了两种不同类型的不确定性，即状态分布概率已知的“风险”和状态分布概率未知的“模糊性”。大量行为实验研究表明，人们在风险场景和模糊场景下的行为模式存在显著的差异。关于风险决策和模糊性决策的区别最早始于 Ellsberg(1961)的研究，Ellsberg 进行了下面的实验，要求被试在明确知道概率的情境和不知道概率的情境二者之间进行选择，发现被试严格偏好于已知概率的决策问题。这一研究结果违背了 Savage 公理，否认了主观概率的存在性，被后人称为 Ellsberg 悖论，从而将风险决策和模糊性决策区别开来。

近年来，一些学者依靠神经科学的工具，表明大脑中可能存在一个神经回路或者大脑的某一特定区域决定了不确定条件下人们的决策行为。在神经金融中，研究者们通过各种不同的实验范式，如表 15.2 中的 Rogers'risk task(Rogers 等，1999；Fecteau 等，2007a)、balloon analog risk task(BART；Lejuez 等，2002；Fecteau 等，2007b)、赌博实验(Hsu 等，2005)和彩票实验(Holt and Laury，2002；Ye 等，2015)，来研究大脑具体的某一区域与风险态度的关系。

表 15.2 风险态度实验简介

实验范式	简介
Rogers'risk task	被试面前有 6 个箱子，箱子有红色和蓝色，红色箱子和蓝色箱子数量可能是 5∶1，4∶2，3∶3。实验进行 n 轮，每一轮被试要求选出最有可能含有“胜利令牌”(winning token)的箱子的颜色。而“胜利令牌”在每个箱子里的可能性是相同的。因此，每一轮中红色箱子和蓝色箱子的个数之比就决定了猜中“胜利令牌”的可能性，即代表风险程度。如果猜中了，被试将会获得奖励；如果猜错了，被试也要承担损失。奖励的数额可能是 90∶10，80∶20，70∶30，60∶40，被试可以看到奖励数额。最高的奖励往往连接的是更高风险的选择。例如，一轮中是 5 个蓝色箱子、1 个红色箱子，选择蓝色猜中的可能性更高，但是蓝色对应的奖励的数额会更小。而如果被试猜错了，损失的数额跟奖励的数额一样。

续表

实验范式	简介
balloon analog risk task	被试通过推入泵(pump)来充气球,气球随时可能爆炸。被试在每一次推入泵的时候决定继续推入泵充气球还是停止。被试收益依据推入的泵的数量计入临时收益账户,一个泵 x 元。如果被试在气球爆炸前停止继续推入泵,则累计泵的价值转入最终收益账户;如果气球在被试推入泵的过程中爆炸,则被试该轮收益清零。风险随着泵的推入而不断增加,而每个气球爆炸的时点不同
赌局实验	每一轮被试需要选择参加赌局,还是选择固定收益。如果被试选择固定收益,则此轮实验结束;如果被试选择参加赌局,即以一定概率获得更高的收益,但也可能没有收益。赌局通过改变概率大小和收益大小,来改变风险程度
彩票实验	实验中有一系列问题,每个问题都由两个不同风险程度(可通过改变概率和收益进行设计)的彩票组成。对于每个问题,被试都需要在两个彩票中作出选择

延伸阅读 15.1　tDCS 刺激前额叶皮质可以降低风险厌恶程度

医学的研究告诉我们,大脑的每个脑区对应着人体不同的功能。那么,大脑什么部位会影响人们在决策时的风险厌恶程度呢?

Ye(2015)等通过 tDCS 设备在被试的背外侧前额叶皮质(DLPFC)分别施加右侧阳极刺激/左侧阴极刺激,左侧阳极刺激/右侧阴极刺激和伪刺激,而被试在刺激前后分别要完成一个内容完全相同但顺序不同的 16 个彩票选择任务。每个彩票选择任务中,被试需要从一个各有 50%机会获得(损失)一定数额 A 或 B 的风险选项与一个各有 50%机会获得(损失)一定数额 C 或 D 的风险选项中作出选择。

实验结果发现,在收益情境下,tDCS 右侧阳极刺激/左侧阴极刺激会使得被试更加风险偏好,然而在损失情境下,tDCS 右侧阳极刺激/左侧阴极刺激会使得被试更加风险厌恶。

资料来源:YE H,CHEN S,HUANG D,et al. *Modulating activity in the prefrontal cortex changes decision-making for risky gains and losses:a transcranial direct current stimulation study*[J]. *Behavioural brain research*,2015,286:17-21.

神经科学领域新兴技术的发展为不确定性下金融决策行为的神经基础研究提供了必要的手段。诸多研究发现,在人们进行不确定性下的风险决策时,大脑的背外侧前额叶皮质和眶额叶皮质(OFC)的活动与人们的风险厌恶程度、模糊性厌恶程度有显著的因果关系。

Hsu(2005)等建立了牌局赌博、常识赌博和信息知晓者对手赌博三个场景,利用功能性核磁共振成像技术研究风险决策和模糊性决策下的脑成像的不同特征,发现"模糊性"条件下人们的决策行为和眶额叶皮质有关。他们的研究从行为学和神经科学双重层面表明风险决策与模糊性决策之间存在显著的不同。Huettel(2006)等发现被试的模糊性偏好也受到前额叶皮质区域的影响,值得一提的是,研究还发现这一区域还与行为冲动性有关,为前额叶皮质为何会影响模糊性偏好提供了可能的解释机制。Knoch(2006)等使用与 tDCS 相类似的经颅磁刺激刺激右侧背外侧前额叶皮质,发现接受这一刺激的被试和没有

接受这一刺激的被试相比，表现出显著的风险寻求的行为特征。Fecteau(2007)等研究了模糊性下的决策行为的神经基础，设计了三个实验组：第一组通过 tDCS 对背外侧前额叶皮质进行左侧阴极/右侧阳极刺激，第二组左侧阳极/右侧阴极刺激，第三组则是伪刺激，并且设置了只进行 tDCS 刺激单侧背外侧前额叶皮质的控制组实验。研究发现接受双边背外侧前额叶皮质刺激的第一组和第二组的决策行为比没有接受刺激的第三组更加的风险保守，而接受单边刺激的控制组模糊性决策行为与没有接受刺激的第三组没有显著区别。

专栏 15.1 风险厌恶和模糊性厌恶

人们在风险场景和模糊性场景下的决策行为模式的差异究竟是源于同一种偏好在不同场景下的表现，还是风险和模糊性本身就是两个完全不同的偏好？为了解决这个问题，我们利用实验经济学方法和经颅直流电刺激(tDCS)设备，分别刺激背外侧前额叶皮质(DLPFC)和眶额叶皮质(OFC)，探究这两个脑区各自与风险、模糊性下决策的因果关系，以检验它们是源于同一种偏好在不同场景下的表现，还是两个完全不同的偏好。

实验任务共 48 个决策任务，其中 24 个风险决策、24 个模糊性决策。每次决策任务开始后，计算机会随机在一个有着红球和蓝球的罐子里抽出一个球。在风险决策中，罐子里的红球个数和蓝球个数均已知。在模糊性决策中，只知道罐子里有几个球，但红球和蓝球各自的个数未知。在每个决策任务中，被试面临两种选择：①选择进行猜球颜色游戏，球的颜色只有红色和蓝色。若猜的球的颜色与计算机抽取的球的颜色相同，可以获得一个高的收益，反之则收益为 0。②选择低的固定收益。如果被试选择 1，则进入后续选择球的颜色；如果被试选择 2，则该决策任务完成，进入下一个决策任务。每个决策任务完成后，不告知被试是否猜对了球的颜色，实验的报酬是出场费加上 48 个决策任务中随机挑选的两轮实验收益之和。在经颅直流电刺激(tDCS)刺激前后，被试均需要完成一次实验任务。

实验结果表明，背外侧前额叶皮质(DLPFC)的活动仅会影响被试在风险场景下的决策行为，而眶额叶皮质(OFC)的活动仅会影响被试在模糊性场景下的决策行为。这一结果在一定程度上支持了风险厌恶和模糊性厌恶可能本身便是两种不同的偏好。

资料来源：杨晓兰等. *Modulating the activity of the DLPFC and OFC has distinct effects on risk and ambiguity decision-making: a tDCS study*[J]. *Frontiers in psychology*, 2017, 8: 1417.

专栏 15.2 为什么通常年轻人爱冒风险，老年人却很保守？来自神经科学的证据

在现实生活中，我们常常会有这样的深刻感受，年轻人总是更爱冒风险，即使面对极大的不确定性，仍然愿意"试一试"，而年纪较大的人却偏保守，更喜欢稳妥的结果，不轻易冒险。事实上，大量的实证研究已经发现人们的风险容忍程度随着年龄的增长而降低。为什么不同年龄的个体会有明显的风险态度差异？

Grubb(2016)等让 52 位 18～88 岁的成年人在一个固定收益选项(得到 5 美元)和一个不确定收益选项(可能得到 5～120 美元，概率未知)之间作出选择。结果和经验发现类似，他们发现年纪大的被试比年轻的被试更多选择固定收益选项，即更加风险厌恶，而且被试的风险厌恶程度随着年龄的增长而增长。到底是什么变量决定了这种风险态度变化

呢？他们将这些数据输入一个模型，结果发现这主要是由右后顶叶皮层的灰质数量驱动的，而不是年龄。根据相关的数据，当人们在做风险决策时，大脑右后顶叶皮层处于活跃状态，而这一脑区的灰质数量与风险偏好相关。随着年龄增长，人们倾向于做更少的风险决策，但是这与因年龄增长而增加的智慧相关，还是和大脑结构相关，一直不为人所知。这些发现表明，在正常衰老过程中，大脑发生的变化对人类决策模式和风险态度的影响可能大于此前预期。

资料来源：GRUBB M A，TYMULA A，GILAIEDOTANS，et al. *Neuroanatomy accounts for age-related changes in risk preferences*[J]. *Nature communications*，2016，7：13822.

15.3 情绪相关的金融决策

情绪会影响人们的金融决策行为，但由于情绪很难被定量测量，传统金融学实验研究难以将其纳入考虑范围。但行为金融学一直将情绪作为影响人们决策的重要变量。正如引导案例中提到的研究，神经金融的研究工具可以在一定程度上定量测量各种情绪，在很大程度上开拓了情绪相关的金融决策研究领域。

1. 后悔

后悔理论认为当决策者发现其他可选择的结果优于自己选择的结果时，会产生后悔情绪(Bell，1983；Loomes 和 Sugden，1982、1987)。而后悔在人们的决策过程中扮演了重要的角色。如果被试预期到某种决策会导致将来的后悔，他们就会避免选择这些选项以使得自己的后悔最小化。后悔也往往会导致更加谨慎的选择。神经科学研究证实了后悔情绪影响决策的理论假设，并进一步探讨了其对应的神经机制。

神经影像学的研究发现，眶额叶皮层、背侧扣带前回(anterior cingulate cortex)、内侧前额叶(medial prefrontal cortex)以及海马(hippocampus)前部等，均会在被试体验到后悔情绪时被激活。

Camille(2004)等在《科学》杂志上发表了一篇关于眶额叶皮层在后悔情绪体验中的作用。他们设计了一种简单的转盘赌博任务，每个转盘由黑色和灰色两种颜色组成，涂色部分的大小代表两种颜色各自的概率，两种颜色各自代表一定的金额。他们分别让健康被试和眶额叶皮层受损被试选择两个具有不同风险概率的转盘中的一个。当被试选定其中一个转盘后，转盘上的箭头会开始旋转，箭头最终停留在哪个颜色，被试就可以获得该颜色对应的金额。同时，被试在每次转盘赌博任务中都要通过回答量表来报告自己的情绪。实验结果发现，当被试得知自己选择的转盘收益不如另外一个放弃的转盘的收益时，健康被试会体验到较为强烈的后悔情绪，而眶额叶皮层受损被试则不会产生后悔情绪，具体表现为皮肤电反应(skin conductance response，SCR)的缺失。由此可见，眶额叶皮层在后悔情绪中起到了重要的作用。关于后悔的众多神经经济学研究从神经水平支持了后悔理论，使得后悔理论有了更坚实的神经学基础，而不再仅仅是理论上的合理解释。

2. 共情

共情由美国著名心理学家罗杰斯提出，指体验别人内心世界的能力，换句话说，对别

人的情绪、思维等的理解以及对别人行为的推测，也就是俗话说的“感同身受”。Gladstein 在 1983 年提出共情的两成分理论，指出共情分为情绪共情和认知共情。情绪共情，指的是对他人情绪的一种感受。脑成像的相关实验结果认为，他人的情绪可以激发自身与这种情绪相关脑区的活动，从而将他人的情感表征转化成个体自身的情感表征，因此使得个体对他人的情绪“感同身受”。因此，情绪共情的神经活动与诱发共情的情绪性质有关。而认知共情，指的是对他人目的、企图、信仰的理解。情绪共情可以产生利他行为的动机，认知共情则有助于选择最合适的办法去帮助别人。因此，共情这一情绪对于社会和谐、构建情感纽带有着一定的意义。

Singer(2004)等将感受他人痛苦的能力称为共情，对配偶进行电击，发现女性被试在配偶被电击时，情感加工相关的脑区(双侧前岛叶、前扣带回吻部)和自己被电击时一样被激活，且其共情问卷得分越高，激活的程度越高。

共情不仅在疼痛、公平等情绪中有所体现，在金融决策中，共情同样作用非凡。Mobbs(2009)等研究发现，人们能够对他人得到的金钱奖励感同身受，从而产生愉悦感，而对和自己相似度(similarity)越高的人，这种金钱奖励的共情越强烈。Ma 等(2011)将被试每 3 人分为一组进行风险投资决策(图 15.1)，其中被试 A 与 B 是朋友，A 与 C 不相识。在实验 1 中，被试 A、B、C 都要做投资任务。而在实验 2 中，被试 A 不需要做投资任务，仅 B、C 做投资任务。两个实验中的被试 A 将做脑扫描，通过事件相关电位记录其脑电位的变化。实验发现在 A 自己参与投资的情况下，A 观察朋友 B 和陌生人 C 投资得失时，A 大脑中事件相关脑电位的“反馈相关负波”(feedback related negativity，FRN)(这个负波反映了对得失感知)没有差异。然而，当不让被试 A 参与投资，A 观察到朋友 B 投资失利比看到陌生人 C 投资失利，诱发了更为明显的反馈相关负波。由此可见，共情在金融行为中也不乏多见，然而现实的金融决策更为复杂，社会关系、竞争状态等都会影响其结果。

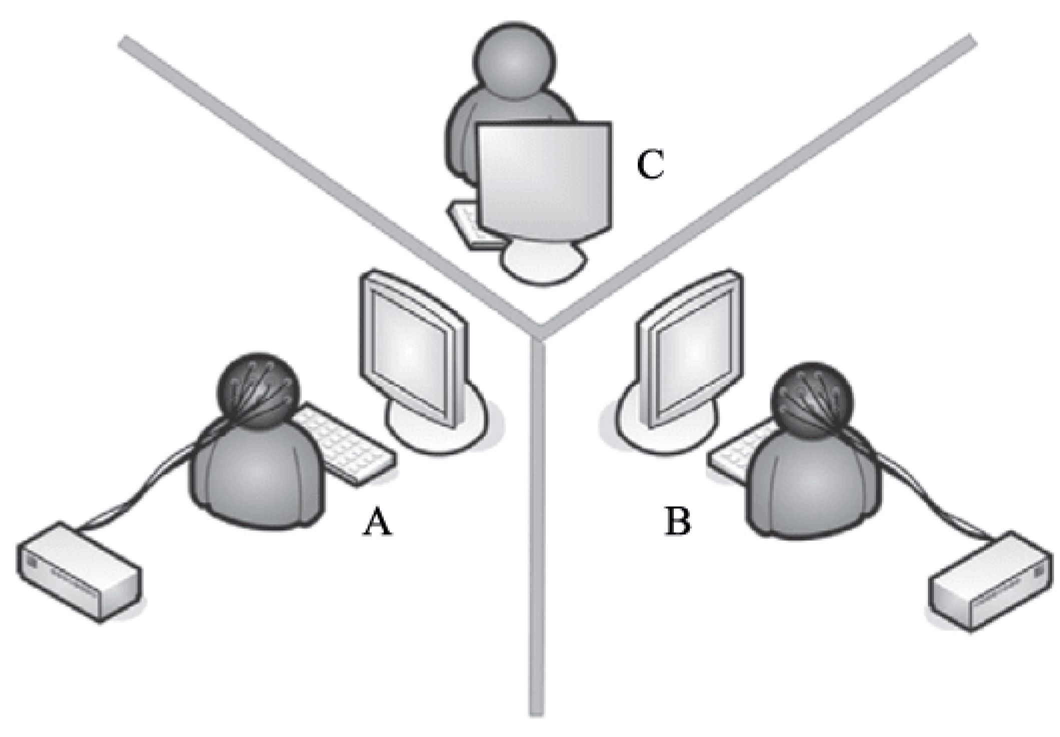


图 15.1　实验示意图

3. 焦虑

个人的焦虑情绪会影响到金融决策。一般而言，相比于低焦虑的人，一个高焦虑的人在作出决策时，往往会受到多方面因素的困扰，难以冷静细致地聚焦问题，而易于作出低效率的决策。试想一下，一个因家庭失败、工作不顺而处于离婚、离职边缘，倍感焦虑的男

性A,和一个家庭和睦,工作业绩出色,没有很大生活压力与事业焦虑的男性B,在面对股市投资时,谁更能够作出理性的决策,谁又易于孤注一掷,作出低效率的决定呢?神经科学已经给出了一定的证据,表明焦虑情绪能够显著地影响到个人的金融决策。

焦虑情绪会影响个人的注意力。Barhami(2005)等在电脑屏幕上分别给高焦虑和低焦虑的人展示包含生气、害怕、悲伤、快乐等情绪以及面无表情的脸,并要求参与者集中注意力于这张面孔图案上,以最快的速度指出随机出现在面孔上下左右任意方位的目标形状。结果显示,高焦虑的人对面孔上出现的目标形状反应更慢。基于ERP波形的分析发现,处于高焦虑中的人的身体反应机能并无大的改变,但其大脑活动更容易被面孔情绪特征所干扰,因而产生低效率的行为表现。

大量神经科学研究已经发现,焦虑情绪会深刻地影响到大脑的活动,甚至影响到整个大脑进行决策的最终过程。Li(2015)等对于焦虑症患者的临床研究发现,焦虑患者的前额皮质(prefrontal cortices)会变得异常活跃,由此给出了焦虑影响个人决策的生理学解释。活跃的前额皮层阻碍了人们去集中注意力,也影响了人们对自身表现进行评估的效率。反映在金融决策领域,焦虑的情绪下,投资者无法集中精力去思考投资策略,同时也无法很好地基于外部信息去及时调整自身的决策行为,从而影响其进行金融决策的效率与正确性。因此,某种程度上来说,焦虑是金融投资者的头号敌人。克服焦虑,舒缓压力,是每个金融投资者必谙的要义。

4. 其他

诸多情绪都会影响人们的金融决策行为,如同情、愤怒、高兴、愧疚、恐慌等。当对某些人表现出同情的时候,可能就不会单纯地以自己的利益为目的,而倾向于帮助他人。如果很高兴,可能会倾向于相信期望发生的事件更容易发生,从而使得期望的事件发生的可能性比不处于高兴状态时更高。在积极情绪中,可能会低估风险的程度,而在消极情绪中,可能会高估风险的程度。

15.4 神经金融学的展望

作为建立在金融学、心理学和神经科学这三大学科基础之上的一门新兴交叉学科,神经金融学方兴未艾。目前,对于金融决策的研究正在逐步深入,从完全理性的传统经济学,到考虑个体心理和情绪等异质性因素的行为金融,再发展至生理层面,神经金融学有着广阔的发展前景。此外,诸多缺乏相关神经金融实验的金融学研究,有着巨大的发展空间。

神经金融学伴随着神经科学工具的发展迅速崛起,通过大脑的活动来解释、预测金融决策行为,成为金融决策理论研究的重要组成部分,对于进一步剖析金融决策的神经决定机制有一定的意义,同时也有助于完善现有的金融决策模型。相对于行为金融学,神经金融学的优势之一是可以为情绪等异质性且难以捕捉记录的因素提供神经证据。神经金融学研究既可以在一定程度上验证已有的金融学理论,同时也为建立新的、更完善的金融学理论奠定了神经学基础。

神经金融学研究也有一定的实际应用价值，如有助于治疗赌博上瘾、冲动购物等临床症状，也为在商业交往中建立良好合作关系以及制定更有效的奖惩措施以提高人们的工作效率等发挥重要的指导作用。

然而，神经科学研究本身也有其局限性。例如，脑成像的 fMRI 技术，只是通过血氧浓度等指标作为神经活动的间接衡量指标，无法直接观察脑区的活动。更重要的是，神经科学研究是一种相关研究而不是因果性研究，大脑的活动模式与特定的心理功能和行为之间的高度相关并不表示二者必然具有因果联系。经颅直流电刺激和经颅磁刺激技术可以研究特定大脑区域与相应金融决策行为之间的因果关系，但它们只对大脑的皮层结构起作用，无法到达更深层的大脑结构，使得这些研究手段应用有所限制。此外，神经科学研究结果解释没有严格的规范，如何科学且合理地对其进行解释仍然是神经金融学目前面临的障碍。尽管已有大量的相关文献发表，但很多研究结果仍然有待日后进一步的实验重复验证。神经科学研究对实验环境和实验操作技术要求较高，环境的影响以及实验操作的细节都有可能会影响实验的结果，且由于神经科学研究成本较高，尤其是功能性磁共振成像，被试样本量均较小，结果存在一定的偶然性。

课程实验指引

欢迎参加实验！如果你在实验说明所描述的规则下认真进行了决策，你将在实验中得到收入。实验结束后，你的实验所得是 20 元的参与费和你在实验中的收益加总。

在本次实验中，你总共需要进行 48 次决策，左上角的数字是决策的题目号，决策的顺序由计算机随机打乱。

每次决策，你都可以选择进行猜球游戏或者选择固定收益。若你选择猜球游戏，请在决策时单击“左边”选项，然后选择“红球”或者“蓝球”，则计算机将根据随机概率抽取一球，再进入下一轮决策。若你选择固定收益，请在决策时单击“右边”选项，则进入下一轮决策。

实验界面中，红色圆表示红球，蓝色圆表示蓝球。圆上方的数字表示红球和蓝球的数量。每个圆正上方的数字分别表示该颜色球的数量。若只出现一个数字在两个圆的中间，则表示红球和蓝球数量的总和。

圆下方的数字表示点数。

当你选择进行猜球游戏时：若点数为正，且你选择的颜色与计算机抽取的颜色相同，你将获得对应点数的筹码，否则你的收益为 0。

当你选择固定收益时：你将 100％获得对应点数的筹码。

图 15.2 表示红球和蓝球一共有 15 个，但是红球和蓝球的具体数目不知道。

若你选择“左边”选项，你需要选择球的颜色。如果你选择的颜色和计算机随机抽取的球的颜色是相同的，则你将获得 24 点筹码，否则你的收益将为 0。

若你选择“右边”选项，你将 100％获得 9 点筹码。



(a) (b) (c)

图 15.2 实验界面

本实验是一个风险态度测试实验，可借助神精科学仪器，研究风险偏好的神经机制。

本章小结

（1）神经金融学是一门将金融学和神经科学与心理学相结合而形成的新兴交叉学科。它通过神经科学的工具来研究金融市场中的决策行为，从基础的生理层面挖掘对决策有重要影响的生理学变量，从神经学层面为人的决策行为提出佐证。

（2）在人们进行不确定性下的金融决策时，大脑的背外侧前额叶皮质和眶额叶皮质的活动与人们的风险厌恶程度、模糊性厌恶程度有显著的因果关系。

（3）情绪会影响人们的金融决策行为。神经金融的研究工具可以在一定程度上定量测量各种情绪，在很大程度上开拓了情绪相关的金融决策研究领域。

（4）目前，对于金融决策的研究正在逐步深入，从完全理性的传统经济学，到考虑个体心理和情绪等异质性因素的行为金融，再发展至生理层面，神经金融学有着广阔的发展前景。

思 考 题

1. 神经金融学主要使用的神经科学设备有哪些？它们各自的作用原理是什么？

2. 哪些情绪会影响人们的金融决策行为？请举一些例子，说明这些情绪将会怎样影响行为。

3. 神经金融学有哪些现实的应用领域？请举例说明。

附录

神经金融研究仪器

主要参考文献

[1] 阿科特,迪弗斯.行为金融学:心理、决策和市场[M].北京:机械工业出版社,2012.

[2] 阿伦森.社会性动物[M].郑日昌,等译.北京:新华出版社,2001.

[3] 巴克.社会心理学[M].天津:南开大学出版社,1986.

[4] 陈国进,吴锋.代理投资、金融危机与金融制度结构[J].金融研究,2002(8):61-66.

[5] 陈昆玉,王跃堂.行为公司金融理论发展评介——源起、心理学基础及研究进展[J].生产力研究,2007,12:139-140,146.

[6] 崔巍.行为公司金融:理论研究及在我国的实践意义[J].中国软科学,2010,4:128-134.

[7] 邓建平,曾勇.上市公司股票更名效应的实证研究[J].闽南师范大学学报(哲学社会科学版),2006,20(3):1-6.

[8] 邓雪莉.互联网金融市场中的群体行为与金融泡沫——基于行为金融学的剖析[J].商情,2015:31.

[9] 丁荣余,张兵.中国证券市场星期效应逐渐消失的经验证据[J].管理工程学报,2005,19(3):145-150.

[10] 董大勇,肖作平.证券信息交流家乡偏误及其对股票价格的影响:来自股票论坛的证据[J].管理世界,2011(1):52-61.

[11] 樊重俊,刘臣,霍良安.大数据分析与应用[M].上海:立信会计出版社,2016.

[12] 奉立城.中国股票市场的"周内效应"[J].经济研究,2000(11):50-57.

[13] 管涛,邓海清,韩会师.泡沫的终结——透视中国股市异动[M].北京:中信出版社,2016.

[14] 韩国文,任伟红.我国封闭式基金折价的实证分析及行为金融学解释[J].中国软科学,2004(4):59-64.

[15] 何兴强,李涛.不同市场态势下股票市场的非对称反应——基于中国上证股市的实证分析[J].金融研究,2007(8a):131-140.

[16] 花贵如,刘志远,许骞.投资者情绪、管理者乐观主义与企业投资行为[J].金融研究,2011(9):178-191.

[17] 黄健青,陈欢,李大夜.基于顾客价值视角的众筹项目成功影响因素研究[J].中国软科学,2015(6):116-127.

[18] 黄玲,周勤.创意众筹的异质性融资激励与自反馈机制设计研究——以"点名时间"为例[J].中国工业经济,2014(6):135-147.

[19] 吉戈伦尔,托德,ABC 研究组.简捷启发式[M].刘永芳,译.上海:华东师范大学出版社,2002.

[20] [法]古斯塔夫・勒庞.乌合之众:大众心理研究[M].北京:中央编译出版社,2004.

[21] 姜付秀,张敏,陆正飞,等.管理者过度自信、企业扩张与财务困境[J].经济研究,2009(1):131-143.

[22] 蒋金锋.我国股票市场指数效应的实证研究[D].长沙:中南大学,2008.

[23] 蒋玉梅,王明照.投资者情绪与股票收益:总体效应与横截面效应的实证研究[J].南开管理评论,2010,13(3):150-160.

[24] 金雪军,杨晓兰.证券市场泡沫实验研究综述[J].国外社会科学,2004(3):20-25.

[25] 兰垠.公司行为金融学:管理者的非理性——一个综述[J].世界经济情况,2007,3:71-75.

[26] 李爱梅,凌文辁.心理账户:理论与应用启示[J].心理科学进展,2007,15(5):727-734.

[27] 李广子,唐国正,刘力.股票名称与股票价格非理性联动——中国 A 股市场的研究[J].管理世界,2011(1):40-51.

[28] 李涛,郭杰.风险态度与股票投资[J].经济研究,2009(2):56-67.

[29] 李涛.社会互动、信任与股市参与[J].经济研究,2006(1):34-45.

[30] 李涛.社会互动与投资选择[J].经济研究,2006(8):45-57.

[31] 李志生,陈晨,林秉旋.卖空机制提高了中国股票市场的定价效率吗?——基于自然实验的证据[J].经济研究,2015(4):165-177.

[32] 廖理,吉霖,张伟强.语言可信吗?借贷市场上语言的作用——来自 P2P 平台的证据[J].清华大学学报,2015(4):413-421.

[33] 林慧婷,王茂林.管理者过度自信、创新投入与企业价值[J].经济管理,2014(11):94-102.

[34] 林立,张菡.公司并购决策研究综述——基于行为公司金融角度[J].现代商贸工业,2009,15:154-155.

[35] 刘斌才.上海证券市场"指数效应"的实证检验[J].对外经贸,2004(4):29-30.

[36] 刘力,田雅静.没有信息,也有反应:中国 A 股市场股票名称变更事件的市场反应研究[J].世界经济,2004(1):44-50.

[37] 刘维奇,刘新新.个人和机构投资者情绪与股票收益——基于上证 A 股市场的研究[J].管理科学学报,2014,17(3):70-87.

[38] 陆磊,刘思峰.中国股票市场具有"节日效应"吗?[J].金融研究,2008(2):127-139.

[39] 路磊,黄京志,吴博.基金排名变化和羊群效应变化[J].金融研究,2014(9):177-191.

[40] 潘国陵.股市泡沫研究[J].金融研究,2000(7):71-79.

[41] 彭叠峰,饶育蕾,王建新.有限注意、投资者行为与资产定价——一个研究评述[J].中南大学学报(社会科学版),2012,18(3):116-122.

[42] 饶育蕾,彭叠峰,成大超.媒体注意力会引起股票的异常收益吗?——来自中国股票市场的经验证据[J].系统工程理论与实践,2010,30(2):287-297.

[43] 饶育蕾,盛虎.行为金融学[M].北京:机械工业出版社,2010.

[44] 沈艺峰,吴世农.我国证券市场过度反应了吗?[J].经济研究,1999(2):21-26.

[45] 施莱费尔.并非有效的市场[M].北京:中国人民大学出版社,2003.

[46] 石荣丽,阳浙江.中国证券投资基金"羊群行为"实证研究[J].经济论坛,2004(7):93-94.

[47] 宋军,吴冲锋.基于分散度的金融市场的羊群行为研究[J].经济研究,2001(11):21-27.

[48] 宋双杰,曹晖,杨坤.投资者关注与 IPO 异象——来自网络搜索量的经验证据[J].经济研究,2011(s1):145-155.

[49] 孙光国,赵健宇.产权性质差异、管理层过度自信与会计稳健性[J].会计研究,2014(5):52-58.

[50] 孙力强,吕鹏.短视损失厌恶的行为金融理论简评[J].中南财经政法大学学报,2008(1):81-85.

[51] 孙培源,施东晖.基于 CAPM 的中国股市羊群行为研究[J].经济研究,2002(2):64-70.

[52] 谭洪涛,蔡利,蔡春.公允价值与股市过度反应——来自中国证券市场的经验证据[J].经济研究,2011(7):130-143.

[53] 唐国豪,姜富伟,张定胜.金融市场文本情绪研究进展[J].经济学动态,2016(11):137-147.

[54] 汪炜,周宇.中国股市"规模效应"和"时间效应"的实证分析——以上海股票市场为例[J].经济研究,2002(10):16-21,30,94.

[55] 王晓翌,陈乾坤.行为金融的本地偏好理论研究综述[J].中南财经政法大学学报,2011(1):50-55.

[56] 王正位,向佳,廖理,等.互联网金融环境下投资者学习行为的经济学分析[J].数量经济技术经济研

究,2016(3):95-111.

[57] 威廉·曼彻斯特.光荣与梦想:1932—1972年美国社会实录:上[M].海口:海南出版社,2004.

[58] 魏星集,夏维力,孙彤彤.基于BW模型的A股市场投资者情绪测度研究[J].管理观察,2014(33):71-73.

[59] 吴佳哲.基于羊群效应的P2P网络借贷模式研究[J].国际金融研究,2015(11):88-96.

[60] 吴世农,许年行,蔡海洪,等.股市泡沫的生成机理和度量[J].财经科学,2002(4):6-11.

[61] 伍岩然,韩立岩.不完全理性、投资者情绪与封闭式基金之谜[J].经济研究,2007(3):117-129.

[62] 希勒.非理性繁荣[M].北京:中国人民大学出版社,2001.

[63] 辛自强.经济心理学经典与前沿实验[M].北京:北京师范大学出版社,2014.

[64] 徐爱农.中国股票市场泡沫测度及其合理性研究[J].财经理论与实践,2007,28(1):34-39.

[65] 徐龙炳,陈历轶.股票送转与管理者双重迎合[J].金融研究,2018(5).

[66] 徐巍,陈冬华.自媒体披露的信息作用——来自新浪微博的实证证据[J].金融研究,2016(3):157-173.

[67] 杨晓兰,高媚,朱淋.社会互动对股票市场的影响——基于新浪财经博客的实证分析[J].证券市场导报,2016(7):50-58.

[68] 杨晓兰,金雪军.我国股票市场熔断机制的磁力效应:基于自然实验的证据[J].金融研究,2017(9):161-177.

[69] 杨晓兰,沈翰彬,祝宇.本地偏好、投资者情绪与股票收益率:来自网络论坛的经验证据[J].金融研究,2016(12):143-158.

[70] 易志高,茅宁.中国股市投资者情绪测量研究:CICSI的构建[J].金融研究,2009(11):174-184.

[71] 游家兴,吴静.沉默的螺旋:媒体情绪与资产误定价[J].经济研究,2012(7):141-152.

[72] 游家兴.谁过度反应,谁反应不足——投资者异质性与收益时间可预测性分析[J].金融研究,2008(4):161-173.

[73] 余明桂,李文贵,潘红波.管理者过度自信与企业风险承担[J].金融研究,2013(1):149-163.

[74] 余明桂,夏新平,邹振松.管理者过度自信与企业激进负债行为[J].管理世界,2006(8):104-112.

[75] 俞庆进,张兵.投资者有限关注与股票收益——以百度指数作为关注度的一项实证研究[J].金融研究,2012(8):152-165.

[76] 曾建光.网络安全风险感知与互联网金融的资产定价[J].经济研究,2015(7):131-145.

[77] 张圣平,于丽峰,李怡宗.媒体报导与中国A股市场盈余惯性——投资者有限注意的视角[J].金融研究,2014(7):154-170.

[78] 张万军.基于大数据的个人信用风险评估模型研究[D].北京:对外经济贸易大学,2015.

[79] 赵静梅,吴风云.数字崇拜下的金融资产价格异象[J].经济研究,2009(6):129-141.

[80] 郑海超,黄宇梦,王涛,等.创新项目股权众筹融资绩效的影响因素研究[J].中国软科学,2015(1):130-138.

[81] 周爱民,孟庆斌.大鱼如何吃小鱼:股市价格泡沫的度量与理性扩容速度的行为金融学分析[M].厦门:厦门大学出版社,2009.

[82] 朱莉琪,皇甫刚.不确定情境中的决策心理——适应与认知[J].心理科学进展,2003,11(5):481-485.

[83] 宗计川,付嘉,包特.交易者认知能力与金融资产价格泡沫:一个实验研究[J].世界经济,2017,40(6):167-192.

[84] ABREU M,MENDES V. *Information,overconfidence and trading:do the sources of information*

matter? [J]. *Journal of economic psychology*, 2012, 33(4): 868-881.

[85] ACEMOGLU D, ZILIBOTTI F. *Was prometheus unbound by chance? Risk, diversification, and growth*[J]. *Journal of political economy*, 1997, 105(4): 709-751.

[86] AGGARWAL R, RIVOLI P. *Seasonal and day-of-the-week effects in four emerging stock markets*[J]. *The financial review*, 1989, 24(4): 541-550.

[87] AGRAWAL A, CATALINI C, GOLDFARB A. *Crowdfunding: geography, social networks, and the timing of investment decisions*[J]. *Journal of economics & management strategy*, 2015, 24(2): 253-274.

[88] ANDERSON L R, HOLT C A. *Information cascades in the laboratory*[J]. *American economic review*, 1997, 87(5): 847-862.

[89] ANTWEILER W, FRANK M Z. *Is all that talk just noise? The information content of internet stock message boards*[J]. *The journal of finance*, 2004, 59(3): 1259-1294.

[90] BAKER M, STEIN J, WURGLER J. *When does the market matter? Stock prices and the investment of equity-dependent firms*[J]. *Quarterly journal of economics*, 2003, 118: 969-1006.

[91] BAKER M, STEIN J. *Market liquidity as a sentiment indicator*[J]. *Journal of financial markets*, 2004, 7(3): 271-299.

[92] BAKER MALCOLM, WURGLER J. *Market timing and capital structure*[J]. *Journal of finance*, 2002, 57: 1-32.

[93] BAKER M P, RUBACK R S, WURGLER J. *Behavioral corporate finance: a survey*[J/OL]. *Social science research network*, 2004.

[94] BAKER M P, WURGLER J. *Investor sentiment and the cross-section of stock returns*[J]. *Journal of finance*, 2006, 61(4): 1645-1680.

[95] BALL R, BROWN P. *An empirical evaluation of accounting income numbers*[J]. *Journal of accounting research*, 1968, 6(2): 159-178.

[96] BANZ R W. *The relationship between return and market value of common stocks*[J]. *Journal of financial economics*, 1981, 9(1): 3-18.

[97] BARBER B M, ODEAN T. *All that glitters: the effect of attention and news on the buying behavior of individual and institutional investors*[J]. *Review of financial studies*, 2008, 21(2): 785-818.

[98] BARBER B M, ODEAN T. *Boys will be boys: gender, overconfidence, and common stock investment*[J]. *Quarterly journal of economics*, 2001, 116(1): 261-292.

[99] BARBER B M, ODEAN T. *Trading is hazardous to your wealth: the common stock investment performance of individual investors*[J]. *Journal of finance*, 2000, 55(2): 773-806.

[100] BARBERIS N, SHLEIFER A, VISHNY R W. *A Model of investor sentiment*[J]. *Journal of financial economics*, 1998, 49(3): 307-343.

[101] BARHAIM Y, LAMY D, GLICKMAN S. *Attentional bias in anxiety: a behavioral and ERP study*[J]. *Brain and cognition*, 2005, 59(1): 11-22.

[102] BELL D E. *Risk premiums for decision regret*[J]. *Management science*, 1983, 29(10): 1156-1166.

[103] BELLEFLAMME P, LAMBERT T, SCHWIENBACHER A. *Crowdfunding: tapping the right crowd*[J]. *Journal of business venturing*, 2014, 29(5): 585-609.

[104] BENARTZI S, THALER R H. *Myopic loss aversion and the equity premium puzzle*[J]. *Quarterly journal of economics*, 1995, 110(1): 73-92.

[105] BEN-DAVID I, GRAHAM J R, HARVEY C R. *Managerial overconfidence and corporate policies* [J/OL]. NBER Working Paper No. 13711, 2007.

[106] BENREPHAEL A, KANDEL S, WOHL A. *Measuring investor sentiment with mutual fund flows* [J]. *Journal of financial economics*, 2012, 104(2): 363-382.

[107] BIAIS B, HILTON D, MAZURIER K, et al. *Judgemental overconfidence, self-monitoring, and trading performance in an experimental financial market*[J]. *Review of economic studies*, 2005, 72(2): 287-312.

[108] BIKHCHANDANI S, HIRSHLEIFER D A, WELCH I. *A theory of fads, fashion, custom, and cultural change as informational cascades*[J]. *Journal of political economy*, 1992, 100(5): 992-1026.

[109] BIKHCHANDANI S, SHARMA S. *Herd behavior in financial markets*[J]. *IMF staff papers*, 2000, 47(3): 279-310.

[110] BOLLEN J, MAO H, ZENG X. *Twitter mood predicts the stock market*[J]. *Journal of computational science*, 2011, 2(1): 1-8.

[111] BOUDREAUX K J. *Discounts and premiums on closed-end mutual funds: a study of valuation* [J]. *Journal of finance*, 1973, 28(2): 515-522.

[112] BREEDEN D T. *An intertemporal asset pricing model with stochastic consumption and investment opportunities*[J]. *Journal of financial economics*, 1979, 7(3): 265-296.

[113] BRENNER M, IZHAKIAN Y, SADE O. *Ambiguity and overconfidence*[J/OL]. SSPN. Working Papers, NO. 1773663, 2011.

[114] BROWN G W, CLIFF M T. *Investor sentiment and asset valuation*[J]. *The journal of business*, 2005, 78(2): 405-440.

[115] BROWN N C, WEI K D, WERMERS R. *Analyst recommendations, mutual fund herding, and overreaction in stock prices*[J]. *Management science*, 2014, 60(1): 1-20.

[116] BURRELL O K. *Possibility of an experimental approach to investment studies*[J]. *Journal of finance*, 1951, 6(2): 211-219.

[117] BURTCH G, GHOSE A, WATTAL S. *Cultural differences and geography as determinants of online prosocial lending*[J]. *Management information systems quarterly*, 2014, 38(3): 773-794.

[118] BURTCH G, GHOSE A, WATTAL S. *The hidden cost of accommodating crowdfunder privacy preferences: a randomized field experiment*[J]. *Management science*, 2015, 61(5): 949-962.

[119] CAI H, CHEN Y, FANG H. *Observational learning: evidence from a randomized natural field experiment*[J]. *American economic review*, 2009, 99(3): 864-882.

[120] CAMERER C F, LOEWENSTEIN G, PRELEC D. *Neuroeconomics: how neuroscience can inform economics*[J]. *Journal of economic literature*, 2005, 43(1): 9-64.

[121] CAMILLE N, CORICELLI G, SALLET J, et al. *The involvement of the orbitofrontal cortex in the experience of regret*[J]. *Science*, 2004, 304(5674): 1167-1170.

[122] CELEN B, KARIV S. *Distinguishing informational cascades from herd behavior in the laboratory*[J]. *The American economic review*, 2004, 94(3): 484-498.

[123] CEYHAN S, SHI X, LESKOVEC J. *Dynamics of bidding in a P2P lending service: effects of herding and predicting loan success*[C]. Proceedings of the 20th international conference on World wide web. ACM, 2011: 547-556.

[124] CHANG E C,CHENG J W,KHORANA A. *An examination of herd behavior in equity markets: an international perspective*[J]. *Journal of banking & finance*,2000,24(10):1651-1679.

[125] CHOPRA N,LAKONISHOK J,RITTER J R. *Measuring abnormal performance:do stocks overreact?* [J]. *Journal of financial economics*,1992,31(2):235-268.

[126] CIPRIANI M,GUARINO A. *Noise trading in a laboratory financial market:a maximum likelihood approach*[J]. *Journal of the european economic association*,2005,3(2/3):315-321.

[127] COLLIER B,HAMPSHIRE R C. *Sending mixed signals:multilevel reputation effects in peer-to-peer lending markets*[J]. *Proceedings of the 2010 ACM conference on computer supported cooperative work*. *ACM*,2010:197-206.

[128] COOPER M J,DIMITROV O,RAU P R. *A rose. com by any other name*[J]. *Journal of finance*,2001,56(6):2371-2388.

[129] COVAL J D,MOSKOWITZ T J. *Home bias at home:local equity preference in domestic portfolios*[J]. *Journal of finance*. 2010,54(6):2045-2073.

[130] CRAINE R. *Rational bubbles:a test*[J]. *Journal of economic dynamics and control*,1993,17(5/6):829-846.

[131] CROSON R,GNEEZY U. *Gender differences in preferences*[J]. *Journal of economic literature*,2009,47(2):448-474.

[132] DANIEL K,HIRSHLEIFER D,SUBRAHMANYAM A. *Investor psychology and security market under-and overreactions*[J]. *Journal of finance*,1998,53(6):1839-1885.

[133] DA Z,ENGELBERG J,GAO P. *In search of attention*[J]. *The journal of finance*,2011,66(5):1461-1499.

[134] DA Z,ENGELBERG J,GAO P. *The sum of all fears:investor sentiment and asset prices*[J]. *Review of financial studies*,2015,28(1):1-32.

[135] DEAVES R,LÜDERS E,LUO G Y. *An experimental test of the impact of overconfidence and gender on trading activity*[J]. *Review of finance*,2010,13(3):555-575.

[136] DE BONDT W F,THALER R H. *Does the stock market overreact*[J]. *Journal of finance*,1985,40(3):793-805.

[137] DE LONG J B,SHLEIFER A,SUMMERS L H,et al. *Noise trader risk in financial markets*[J]. *Journal of political economy*,1990,98(4):703-738.

[138] DESHMUKH S,GOEL A M,HOWE K M. *CEO overconfidence and dividend policy*[J]. *Journal of financial intermediation*,2013,22(3):440-463.

[139] DIBA B T,GROSSMAN H I. *The theory of rational bubbles in stock prices*[J]. *Economic journal*,1988,98(392):746-754.

[140] Dimson,Elroy and Marsh,Paul and Staunton,Mike,The Worldwide Equity Premium:A Smaller Puzzle[C]. Chapter 11 of R Mehra(Ed),Handbook of the Equity Risk Premium,2006.

[141] DOUKAS J A,MILONAS N T. *Investor sentiment and the closed-end fund puzzle:out-of-sample evidence*[J]. *European financial management*,2004,10(2):235-266.

[142] DUARTE J,SIEGEL S,YOUNG L. *Trust and credit:the role of appearance in peer-to-peer lending*[J]. *Review of financial studies*,2012,25(8):2455-2484.

[143] DUFLO E,SAEZ E. *The role of information and social interactions in retirement plan decisions: evidence from a randomized experiment*[J]. *Quarterly journal of economics*,2003,118(3):815-

842.

[144] DURLAUF S N,IOANNIDES Y M. *Social interactions*[J]. *Annu. Rev. Econ.* ,2010,2(1):451-478.

[145] ELLSBERG D. *Risk, Ambiguity, and the Savage Axioms*[J]. *Quarterly journal of economics*, 1961,75(4):643-669.

[146] ENGELBERG J, MANSKI C F, WILLIAMS J. *Comparing the point predictions and subjective probability distributions of professional forecasters*[J]. *Journal of business & economic statistics*,2009,27(1):30-41.

[147] ENGLE R F, YOO B S. *Forecasting and testing in co-integrated systems*[J]. *Journal of econometrics*,1987,35(1):143-159.

[148] EVERETT C R. *Group membership,relationship banking and loan default risk:the case of online social lending*[J]. *Banking and finance review*,2015,7(2).

[149] FAMA E F,FISHER L,JENSEN MC,ROLL R. *The adjustment of stock prices to new information*[J]. *International economic review*,1969,10(1):1-21.

[150] FECTEAU S,KNOCH D,FREGNI F,et al. *Diminishing risk-taking behavior by modulating activity in the prefrontal cortex:a direct current stimulation study*[J]. *The journal of neuroscience*, 2007,27(46):12500-12505.

[151] FECTEAU S,PASCUALLEONE A,ZALD D H,et al. *Activation of prefrontal cortex by transcranial direct current stimulation reduces appetite for risk during ambiguous decision making* [J]. *The journal of neuroscience*,2007,27(23):6212-6218.

[152] FELLNERROHLING G,KRUGEL S. *Judgmental overconfidence and trading activity*[J]. *Journal of economic behavior and organization*,2014(107):827-842.

[153] FISHER K L,STATMAN M. *Investor Sentiment and Stock Returns*[J]. *Financial analysts journal*,2000,56(2):16-23.

[154] FRANK C. *The behavior of stock prices on Fridays and Mondays*[J]. *Financial analysts journal*, 1973,29(6):67-69.

[155] FRENCH K R,POTERBA J M. *Investor diversification and international equity markets*[J]. *The American economic review*,1991,81(2):222-226.

[156] FRENCH K R. *Stock returns and the weekend effect*[J]. *Journal of financial economics*,1980, 8(1):55-69.

[157] FROOT K A,SCHARFSTEIN D S,STEIN J C. *Herd on the street:informational inefficiencies in a market with short-term speculation*[J]. *Journal of finance*,1992,47(4):1461-1484.

[158] FRYDMAN C,CAMERER C F. *Neural evidence of regret and its implications for investor behavior*[J]. *Review of financial studies*,2016,29(11):3108-3139.

[159] GAO Z,REN H,ZHANG B. *Googling investor sentiment around the world*[J]. *Journal of financial and quantifative analysis*,2018(11).

[160] GARBER P M. *Famous first bubbles*[J]. *Journal of economic* perspectives,1990,4(2):35-54.

[161] GARCIA D. *Sentiment during recessions*[J]. *Journal of finance*,2012,68(3):1267-1300.

[162] GERVAIS S,GOLDSTEIN I. *Overconfidence and team coordination*[J/OL], SSRN working paper, No. 470787,2003. Social Science Research Network,2003.

[163] GERVAIS S, HEATON J B, ODEAN T. *Overconfidence, compensation contracts, and capital budgeting*[J]. *Journal of finance*,2011,66(5):1735-1777.

[164] GERVAIS S, HEATON J B, ODEAN T. *The positive role of overconfidence and optimism in investment policy*[J/OL]. SSRN working paper, NO. 361200, 2002.

[165] GERVAIS S, ODEAN T. *Learning to be overconfident*[J]. *Review of financial studies*, 2001, 14(1): 1-27.

[166] GIGERENZER G, HOFFRAGE U, KLEINBÖLTING H. *Probabilistic mental models: a brunswikian theory of confidence*[J]. *Psychological review*, 1991, 98(4): 506.

[167] GLASER M, WEBER M. *Overconfidence and trading volume*[J]. *Geneva risk & insurance review*, 2007, 32(1): 1-36.

[168] GOEL A M, THAKOR A V. *Overconfidence, CEO selection, and corporate governance*[J]. *Journal of finance*, 2008, 63(6): 2737-2784.

[169] GRAHAM J R, Harvey C R. *The theory and practice of corporate finance: Evidence from the field*[J]. *Journal of financial economics*, 2001, 60: 187-243.

[170] GRIFFIN D, TVERSKY A. *Heuristics and biases: the weighing of evidence and the determinants of confidence*[J]. *Cognitive psychology*, 2002, 24(3): 411-435.

[171] GRINBLATT M, KELOHARJU M. *Sensation seeking, overconfidence, and trading activity*[J]. *Journal of finance*, 2009, 64(2): 549-578.

[172] GRINBLATT M, TITMAN S, WERMERS R. *Momentum investment strategies, portfolio performance, and herding: a study of mutual fund behavior*[J]. *American economic review*, 1995, 85(5): 1088-1105.

[173] GRUBB M A, TYMULA A, GILAIEDOTAN S, et al. *Neuroanatomy accounts for age-related changes in risk preferences*[J]. *Nature communications*, 2016, 7: 13822.

[174] GRULLON G, KANATAS G, WESTON J P. *Advertising, Breadth of Ownership, and Liquidity*[J]. *Review of financial studies*, 2004, 17(2): 439-461.

[175] GRULLON G, WANG F A. *Closed-end fund discounts with informed ownership differential*[J]. *Journal of financial intermediation*, 2001, 10(2): 171-205.

[176] HARRIS M, RAVIV A. *Differences of opinion make a horse race*[J]. *Review of financial studies*, 1993, 6(3): 473-506.

[177] HARUVY E, NOUSSAIR C N. *The effect of short selling on bubbles and crashes in experimental spot asset markets*[J]. *Journal of finance*, 2006, 61(3): 1119-1157.

[178] HAYWARD M L, HAMBRICK D C. *Explaining the premiums paid for large acquisitions: evidence of CEO hubris*[J]. *Administrative science quarterly*, 1997, 42(1): 103-127.

[179] HEATON J B. *Managerial optimism and corporate finance*[J]. *Financial management*, 2002, 31: 33-45.

[180] HERZENSTEIN M, DHOLAKIA U M, ANDREWS R L. *Strategic herding behavior in peer-to-peer loan auctions*[J]. *Journal of interactive marketing*, 2011, 25(1): 27-36.

[181] HIRSHLEIFER D. *Limited attention, information disclosure, and financial reporting*[J]. *Journal of accounting and economics*, 2003, 36(1): 337-386.

[182] HIRSHLEIFER D A, LOW A, TEOH S H. *Are overconfident CEOs better innovators*[J]. *Journal of finance*, 2012, 67(4): 1457-1498.

[183] HIRSHLEIFER D A, LUO G Y. *On the survival of overconfident traders in a competitive securities market*[J]. *Journal of financial markets*, 2001, 4(1): 73-84.

[184] HIRSHLEIFER D A, SHUMWAY T. *Good day sunshine: stock returns and the weather*[J]. *Journal of finance*, 2003, 58(3): 1009-1032.

[185] HOLT C A, LAURY S K. *Risk aversion and incentive effects*[J]. *The American economic review*, 2002, 92(5): 1644-1655.

[186] HONG H, WANG JIANG, YU JAILIN. *Firms as buyers of last resort*[J]. *Journal of financial economics*, 2008, 88: 119-145.

[187] HONG H G, KUBIK J D, SOLOMON A. *Security analysts' career concerns and herding of earnings forecasts*[J]. *The RAND journal of economics*, 2000, 31(1): 121-144.

[188] HONG H G, KUBIK J D, STEIN J C. *The only game in town: stock-price consequences of local bias*[J]. *Journal of financial economics*, 2008, 90(1): 20-37.

[189] HONG H G, STEIN J C. *A unified theory of underreaction, momentum trading and overreaction in asset markets*[J]. *Journal of finance*, 1999, 54(6): 2143-2184.

[190] HSU M, BHATT M, ADOLPHS R, et al. *Neural systems responding to degrees of uncertainty in human decision-making*[J]. *Science*, 2005, 310(5754): 1680-1683.

[191] HUANG A H, ZANG A, ZHENG R. *Evidence on the information content of text in analyst reports*[J]. *The accounting review*, 2014, 89(6): 2151-2180.

[192] HUANG RONGBING, RITTER J. *Testing theories of capital structure and estimating the speed of adjustment*[J]. *Journal of financial and quantitative analysis*, 2009, 44: 237-271.

[193] HUCK S, OECHSSLER J. *Informational cascades in the laboratory: do they occur for the right reasons?* [J]. *Journal of economic psychology*, 2000, 21(6): 661-671.

[194] HUETTEL S A, STOWE C J, GORDON E M, et al. *Neural signatures of economic preferences for risk and ambiguity*[J]. *Neuron*, 2006, 49(5): 765-775.

[195] IYER R, KHWAJA A I, LUTTMER E F P, et al. *Screening in new credit markets: can individual lenders infer borrower creditworthiness in Peer-to-Peer lending?* [J/OL]. SSRN working paper, NO. 1570115, 2009.

[196] JEGADEESH N, TITMAN S. *Returns to buying winners and selling losers: implications for stock market efficiency*[J]. *Journal of finance*, 1993, 48(1): 65-91.

[197] JENTER D, LEWELLEN K. *CEO preferences and acquisitions*[J]. *Journal of finance*, 2015, 70(6): 2813-2852.

[198] KAHNEMAN, D. *Attention and effort*[M]. Englewood Cliffs, New Jersey: Prentice-Hall, 1973.

[199] KAHNEMAN D, TVERSKY A. *Choices, values, and frames*[J]. *American psychologist*, 1984, 39(4): 341-350.

[200] KAHNEMAN D, TVERSKY A. *Prospect theory: an analysis of decision under risk*[J]. *Econometrica*, 1979, 47(2): 263-291.

[201] KAHNEMAN D, TVERSKY A. *The simulation heuristic*[C]//KAHNEMAN D, SLOVIC P, TVERSKY A. Judgement Under Uncertainty: Heuristics & Biases, 1982. *Judgement under uncertainty: heuristics and biases*. New York: Cambridge University Press, 1982: 201-208.

[202] KAMSTRA M J, KRAMER L A, LEVI M D. *Losing sleep at the market: the daylight saving anomaly: reply*[J]. *The American economic review*, 2000, 90(4): 1005-1011.

[203] KAUL A, MEHROTRA V, MORCK R. *Demand curves for stocks do slope down: new evidence from an index weights adjustment*[J]. *Journal of finance*, 2000, 55(2): 893-912.

[204] Kazim Hussain and Elsa Prieto, Big Data in the Finance and Insurance Sectors[C]. Chapter 12 of J. M. Cavanillas et al. (eds.), New Horizons for a Data-Driven Economy, Springer International Publishing. 2016.

[205] KEIBER K L. *Managerial compensation contracts and overconfidence*[J/OL]. *Social science research network*, 2002.

[206] KELLY F. *Why you win or lose: The psychology of speculation*[M]. Boston: Houghton Mifflin, 1930.

[207] KEYNES J M. *The general theory of employment, interest and money*[M]. Macmillan Cambridge University Press. Harcourt, Brace. 1936.

[208] KIM M K, BURNIE D A. *The firm size effect and the economic cycle*[J]. *Journal of financial research*, 2002, 25(1): 111-124.

[209] KINDLEBERGER C P. *Manias, panics and crashes: a history of financial crises*[M]. New York: Basic Books, 1978.

[210] KIVETZ R. *Advances in research on mental accounting and reason-based choice*[J]. *Marketing letters*, 1999, 10(3): 249-266.

[211] KLEIDON A W. *Variance bounds tests and stock price valuation models*[J]. *Journal of political economy*, 1986, 94(5): 953-1001.

[212] KNIGHT F H. *Risk, uncertainty, and profit*[M]. Boston: Houghton Mifflin, 1921.

[213] KNOCH D, GIANOTTI L R, PASCUALLEONE A, et al. *Disruption of right prefrontal cortex by low-frequency repetitive transcranial magnetic stimulation induces risk-taking behavior*[J]. *The Journal of neuroscience*, 2006, 26(24): 6469-6472.

[214] KUMAR A, LEE C M. *Retail investor sentiment and return comovements*[J]. *Journal of finance*, 2006, 61(5): 2451-2486.

[215] LAKONISHOK J, SHLEIFER A, VISHNY R W. *The impact of institutional trading on stock prices*[J]. *Journal of financial economics*, 1992, 32(1): 23-43.

[216] LAMBERT J, BESSIERE V, GOALA G N. *Does expertise influence the impact of overconfidence on judgment, valuation and investment decision?* [J]. *Journal of economic psychology*, 2012, 33(6): 1115-1128.

[217] LANGER E J. *The Illusion of Control*[J]. *Journal of personality and social psychology*, 1975, 32(2): 311-328.

[218] LARRICK R P, BURSON K A, SOLL J B. *Social comparison and confidence: when thinking you're better than average predicts overconfidence (and when it does not)*[J]. *Organizational behavior and human decision processes*, 2007, 102(1): 76-94.

[219] LARWOOD L, WHITTAKER W. *Managerial myopia: self-serving biases in organizational planning*[J]. *Journal of applied psychology*, 1977, 62(2): 194-198.

[220] LEE C M, SHLEIFER A, THALER R H. *Investor Sentiment and the Closed-End Fund Puzzle*[J]. *Journal of finance*, 1991, 46(1): 75-109.

[221] LEJUEZ C W, READ J P, KAHLER C W, et al. *Evaluation of a behavioral measure of risk taking: the balloon analogue risk task (BART)*[J]. *Journal of experimental psychology: applied*, 2002, 8(2): 75-84.

[222] LIBERMAN V. *Local and global judgments of confidence*[J]. *Journal of experimental psychol-*

ogy: learning, memory and cognition, 2004, 30(3): 729-732.

[223] LIN M, PRABHALA N R, VISWANATHAN S. *Judging borrowers by the company they keep: friendship networks and information asymmetry in online peer-to-peer lending*[J]. *Management science*, 2013, 59(1): 17-35.

[224] LIN Y, HU S, CHEN M. *Managerial optimism and corporate investment: some empirical evidence from taiwan*[J]. *Pacific-basin finance journal*, 2005, 13(5): 523-546.

[225] LIU D, BRASS D J, LU Y, et al. *Friendships in online peer-to-peer lending: pipes, prisms, and relational herding*[J]. *Management information systems quarterly*, 2015, 39(3): 729-742.

[226] LIU H, COLMAN A M. *Ambiguity aversion in the long run: repeated decisions under risk and uncertainty*[J]. *Journal of economic psychology*, 2009, 30(3): 277-284.

[227] LIU S. *Changes in the nikkei 500: new evidence for downward sloping demand curves for stocks*[J]. *International review of finance*, 2010, 1(4): 245-267.

[228] LIU Y, CHEN R, CHEN Y, et al. "*I loan because...*": *understanding motivations for pro-social lending*[C]. Acm international conference on web search & data mining, 2012: 503-512.

[229] LI Y, WANG W, LIU T, et al. *Source analysis of P3a and P3b components to investigate interaction of depression and anxiety in attentional systems*[J]. *Scientific reports*, 2015, 5(1): 17138.

[230] LOH R K. Investor attention and the under reaction to stock recommendations[J]. *Financial management*, 2010, 39(3): 1223-1252.

[231] LOOMES G, SUGDEN R. *Regret theory: an alternative theory of rational choice under uncertainty*[J]. *The economic journal*, 1982, 92(368): 805-824.

[232] LOOMES G, SUGDEN R. *Some implications of a more general form of regret theory*[J]. *Journal of economic theory*, 1987, 41(2): 270-287.

[233] MALKIEL B G. *The valuation of closed-end investment-company shares*[J]. *Journal of finance*, 1977, 32(3): 847-859.

[234] MALMENDIER U, TATE G A, YAN J. *Overconfidence and early-life experiences: the effect of managerial traits on corporate financial policies*[J]. *Journal of finance*, 2011, 66(5): 1687-1733.

[235] MALMENDIER U, TATE G A, YAN J. *Overconfidence and early-life experiences: the effect of managerial traits on corporate financial policies*[J]. *Journal of finance*, 2011, 66(5): 1687-1733.

[236] MALMENDIER U, TATE G A. *Behavioral CEOs: the role of managerial overconfidence*[J]. *Journal of economic perspectives*, 2015, 29(4): 37-60.

[237] MALMENDIER U, TATE G A. *CEO overconfidence and corporate investment*[J]. *Journal of finance*, 2005, 60(6): 2661-2700.

[238] MALMENDIER U, TATE G A. *Who makes acquisitions? CEO overconfidence and the market's reaction*[J]. *Journal of financial economics*, 2008, 89(1): 20-43.

[239] MA Q, SHEN Q, XU Q, et al. *Empathic responses to others' gains and losses: an electrophysiological investigation*[J]. *NeuroImage*, 2011, 54(3): 2472-2480.

[240] MARCH J G, SHAPIRA Z. *Managerial perspectives on risk and risk taking*[J]. *Management science*, 1987, 33(11): 1404-1418.

[241] MAUG E, NAIK N. *Herding and delegated portfolio management: the impact of relative performance evaluation on asset allocation*[J]. *Quarterly journal of finance*, 2011, 1(2): 265-292.

[242] MCCLURE S M, LAIBSON D, LOEWENSTEIN G, et al. *Separate neural systems value immediate*

and delayed monetary rewards[J]. *Science*, 2004, 306(5695): 503-507.

[243] MCGRATTAN E R, PRESCOTT E C. *Average debt and equity returns: puzzling?* [J]. *The American economic review*, 2003, 93(2): 392-397.

[244] MEHDIAN S, PERRY M J. *The reversal of the monday effect: new evidence from US equity markets*[J]. *Journal of business finance & accounting*, 2001, 28(7/8): 1043-1065.

[245] MEHRA R, PRESCOTT E C. *The equity premium: a puzzle*[J]. *Journal of monetary economics*, 1985, 15(2): 145-161.

[246] MERTON R C. *A simple model of capital market equilibrium with incomplete information*[J]. *Journal of Finance*, 1987, 42(3): 483-510.

[247] MICHELS J. *Do Unverifiable disclosures matter? Evidence from peer-to-peer lending*[J]. *The accounting review*, 2012, 87(4): 1385-1413.

[248] MILLER D T, ROSS M. *Self-Serving Biases in the Attribution of Causality: Fact or Fiction?* [J]. *Psychological bulletin*, 1975, 82(2), 213-225.

[249] MOBBS D, YU R, MEYER M, et al. *A key role for similarity in vicarious reward*[J]. *Science*, 2009, 324(5929): 900.

[250] MOORE D A, HEALY P J. *The Trouble With Overconfidence*[J]. *Psychological review*, 2008, 115(2): 502-517.

[251] NEAL R, WHEATLEY S M. *Do Measures of Investor Sentiment Predict Returns?* [J]. *The Journal of financial and quantitative analysis*, 1998, 33(4): 523-547.

[252] NORD W R, BOTTOM W P. *Heuristics and biases: the psychology of intuitive judgment book* [M]//*The calibration of expert judgement: heuristics and biases beyond the laboratory*, 2004.

[253] Nöth M, Weber, M. *Information aggregation with random ordering: cascades and overconfidence* [J]. *The economic journal*, 2003, 113(484): 166-189.

[254] ODEAN T. *Volume, volatility, price, and profit when all traders are above average*[J]. *Journal of Finance*, 1998, 53(6): 1887-1934.

[255] OLIVER B R. *The impact of management confidence on capital structure*. *Social Science Electronic Publishing*, 2005.

[256] PABLO D A, ANDREW W L. *The wisdom of twitter crowds: predicting stock market reactions to FOMC meetings via twitter feeds*[J]. *Journal of portfolio management*, 2016, 42(5): 123-134.

[257] RABIN M. *Risk aversion and expected-utility theory: a calibration theorem* [J]. *Econometrica*, 2000, 68(5): 1281-1292.

[258] ROGERS R D, OWEN A M, MIDDLETON H, et al. *Choosing between small, likely rewards and large, unlikely rewards activates inferior and orbital prefrontal cortex*[J]. *The journal of neuroscience*, 1999, 19(20): 9029-9038.

[259] ROLL R. *The hubris hypothesis of corporate takeovers*[J]. *The journal of business*, 1986, 59(2): 197-216.

[260] SANDRONI A, SQUINTANI F. *Overconfidence and asymmetric information: the case of insurance*[J]. *Journal of economic behavior and organization*, 2013: 149-165.

[261] SCHARFSTEIN D S, STEIN J C. *Herd behavior and investment*[J]. *The American economic review*, 1990, 80(3): 465-479.

[262] SCHMELING M. *Institutional and individual sentiment: smart money and noise trader risk?* [J].

International journal of forecasting,2007,23(1):127-145.

[263] SCHRAND C M,ZECHMAN S L. *Executive overconfidence and the slippery slope to financial misreporting*[J]. *Journal of accounting and economics*,2012,53(1):311-329.

[264] SCOTT B. *Scientific investment analysis—science or fiction?* [J]. *Financial analysts journal*,1967,23(1):93-97.

[265] SEASHOLES M S,WU G Y J. *Predictable behavior,profits,and attention*[J]. *Journal of empirical finance*,2007,14(5):590-610.

[266] SHILLER R J,FISCHER S,FRIEDMAN B M. *Stock prices and social dynamics*[J]. *Brookings papers on economic activity*,1984(2):457-510.

[267] SHILLER R J. *Do stock prices move too much to be justified by subsequent changes in dividends?*[J]. *American economic review*,1981,71(3):421-436.

[268] SHLEIFER A. *Do demand curves for stocks slope down*[J]. *Journal of finance*,1986,41(3):579-590.

[269] SIECK W R,YATES J F. *Overconfidence effects in category learning:a comparison of connectionist and exemplar memory models*[J]. *Journal of experimental psychology:learning,memory and cognition*,2001,27(4):1003-1021.

[270] SIMON H A. *Rational choice and the structure of the environment*[J]. *Psychological review*,1956,63(2):129-138.

[271] SINGER T,SEYMOUR B,ODOHERTY J P,et al. *Empathy for pain involves the affective but not sensory components of pain*[J]. *Science*,2004,303(5661):1157-1162.

[272] SMITH V L,SUCHANEK G L,WILLIAMS A W. *Bubbles,crashes,and endogenous expectations in experimental spot asset markets*[J]. *Econometrica*,1988,56(5):1119-1151.

[273] SMITH V L. *Rational choice:the contrast between economics and psychology*[J]. *Journal of political economy*,1991,99(4):877-897.

[274] SOLL J B,KLAYMAN J. *Overconfidence in interval estimates*[J]. *Journal of experimental psychology:learning,memory and cognition*,2004,30(2):299-314.

[275] SOLOMON D H,SOLTES E,SOSYURA D. *Winners in the spotlight:media coverage of fund holdings as a driver of flows*[J]. *Journal of financial economics*,2014,113(1):53-72.

[276] STARMER C. *Developments in non-expected utility theory:the hunt fora descriptive theory of choice under risk*[J]. *Journal of economic literature*,2000,38(2):332-382.

[277] STATMAN M,THORLEY S,VORKINK K. *Investor overconfidence and trading volume*[J]. *Review of financial studies*,2006,19(4):1531-1565.

[278] STATMAN M. *Behavioral finance:finance with normal people*[J]. *Borsa Istanbul review*,2014,14(2):65-73.

[279] STEIN J C. *Rational capital budgeting in an irrational world*[J]. *Journal of business*,1996,69:429-455.

[280] STEWART M C,Majluf N S. *Corporate financing and investment decisions when firms have information that investors do not have*[J]. *Journal of financial economics*,1984,13:187-221.

[281] STIGLITZ J E. *Symposium on bubbles*[J]. *The journal of economic perspectives*,1990,4(2):13-18.

[282] STRAHILEVITZ M A,ODEAN T,BARBER B M. *Once burned,twice shy:how naive learning,*

counterfactuals, and regret affect the repurchase of stocks previously sold[J]. *Journal of marketing research*, 2011, 48: 102-120.

[283] SVENSON O. *Are we all less risky and more skillful than our fellow drivers*[J]. *Acta psychologica*, 1981, 47(2): 143-148.

[284] TAYLOR S E, BROWN J D. *Illusion and well-being: a social psychological perspective on mental health*[J]. *Psychological bulletin*, 1988, 103(2): 193-210.

[285] TEIGEN K H, JORGENSEN M. *When 90% confidence intervals are 50% certain: on the credibility of credible intervals*[J]. *Applied cognitive psychology*, 2005, 19(4): 455-475.

[286] THALER R. *Mental accounting matters*[J]. *Journal of behavior decision making*, 1999(12): 183-205.

[287] THALER R H. *Mental accounting and consumer choice*[J]. *Marketing science*, 2008, 27(1): 15-25.

[288] THALER R H. *Toward a positive theory of consumer choice*[J]. *Journal of economic behavior and organization*, 1980, 1(1): 39-60.

[289] TIROLE J. *Asset bubbles and overlapping generations*[J]. *Econometrica*, 1985, 53(6): 1499-1528.

[290] TRINUGROHO I, SEMBEL R. *Overconfidence and excessive trading behavior: an experimental study*[J]. *International journal of biometrics*, 2011, 6(7).

[291] TVERSKY A, KAHNEMAN D. *Advances in prospect theory: cumulative representation of uncertainty*[J]. *Journal of risk and uncertainty*, 1992, 5(4): 297-323.

[292] TVERSKY A, KAHNEMAN D. *Judgment under uncertainty: heuristics and biases*[J]. *Science*, 1974, 185(4157): 1124-1131.

[293] TVERSKY A, KAHNEMAN D. *Rational choice and the framing of decisions*[J]. *The journal of business*, 1986, 59(4): 251-278.

[294] TVERSKY A, KAHNEMAN D. *The framing of decisions and the psychology of choice*[J]. *Science*, 1981, 211(4481): 453-458.

[295] WEBER E U, BLAIS A, BETZ N E. *A domain-specific risk-attitude scale: measuring risk perceptions and risk behaviors*[J]. *Journal of behavioral decision making*, 2002, 15(4): 263-290.

[296] WEIL P. *Confidence and the real value of money in an overlapping generations economy*[J]. *Quarterly journal of economics*, 1987, 102(1): 1-22.

[297] WEINSTEIN N D, KLEIN W M. *Resistance of personal risk perceptions to debiasing interventions*[J]. *Health psychology*, 1995, 14(2): 132-140.

[298] WEINSTEIN N D. *Unrealistic optimism about future life events*[J]. *Journal of personality and social psychology*, 1980, 39(5): 806-820.

[299] WERMERS R. *Mutual fund herding and the impact on stock prices*[J]. *Journal of finance*, 1999, 54(2): 581-622.

[300] XIONG W, YU J. *The Chinese warrants bubble*[J]. *American economic review*, 2011, 101(6): 2723-2753.

[301] YANG X, ZHU L. *Ambiguity vs risk: an experimental study of overconfidence, gender and trading activity*[J]. *Journal of behavioral and experimental finance*, 2016(9): 125-131.

[302] YANG X. *The role of photographs in online peer-to-peer lending behavior*[J]. *Social behavior and personality*, 2014, 42(3): 445-452.

[303] YAO S, LUO D. *The economic psychology of stock market bubbles in China*[J]. *The world econo-*

my,2009,32(5):667-691.

[304] YE H,CHEN S,HUANG D,et al. *Modulating activity in the prefrontal cortex changes decision-making for risky gains and losses:a transcranial direct current stimulation study*[J]. *Behavioural brain research*,2015,286:17-21.

[305] ZAROWIN P. *Does the stock market overreact to corporate earnings information?* [J]. *The journal of finance*,1989,44(5):1385-1399.

[306] ZHANG J,LIU P. *Rational herding in microloan markets*[J]. *Management science*,2012,58(5):892-912.

[307] ZHENG H,LI D,WU J,et al. *The role of multidimensional social capital in crowdfunding:a comparative study in China and US*[J]. *Information & management*,2014,51(4):488-496.

从入门到实战·微课视频

基础知识→核心技术→高级应用→项目实战

Python项目案例开发从入门到实战

爬虫、游戏和机器学习

微课版

郑秋生 夏敏捷 主编



- **零基础入门到实战**

  20个不同类型的完整项目案例

- **全程语音讲解**

  600分钟高品质配套教学视频

- **配套资源丰富**

  提供完整的源码和教学课件

600分钟视频讲解

清華大學出版社

从入门到实战·微课视频

Python项目案例开发从入门到实战

——爬虫、游戏和机器学习

郑秋生　夏敏捷　主编

宋宝卫　李　娟　副主编

清华大学出版社

北　京

内 容 简 介

本书以 Python 3.5 为编程环境，从基本的程序设计思想入手，逐步展开 Python 语言教学，是一本面向广大编程学习者的程序设计类图书。本书以案例带动知识点的讲解，将 Python 知识点分解到各个不同的案例，每个案例各有侧重点，同时展示实际项目的设计思想和设计理念，使读者可以举一反三。

本书案例具有实用性，例如校园网搜索引擎、小小翻译器、抓取百度图片这些爬虫案例略加修改可以应用到实际项目中；还有通过微信通信协议开发微信机器人、机器学习的文本分类、基于卷积神经网络的手写体识别等案例；另外是一些大家耳熟能详的游戏案例，例如连连看、推箱子、中国象棋、网络五子棋、两人麻将、人物拼图和飞机大战等游戏。通过本书，读者将掌握 Python 编程技术和技巧，学会面向对象的设计方法，了解程序设计的所有相关内容。本书不仅为读者列出了完整的代码，同时对所有的源代码都进行了非常详细的解释，通俗易懂、图文并茂。扫描每章提供的二维码可观看知识点的视频讲解。

本书适用于 Python 语言学习者、程序设计人员和游戏编程爱好者。

图书在版编目（CIP）数据

Python 项目案例开发从入门到实战：爬虫、游戏和机器学习 / 郑秋生，夏敏捷主编. —北京：清华大学出版社，2019

（从入门到实战 • 微课视频）

ISBN 978-7-302-45970-5

Ⅰ. ①P…　Ⅱ. ①郑…　②夏…　Ⅲ. ①软件工具-程序设计　Ⅳ. ①TP311.561

中国版本图书馆 CIP 数据核字（2018）第 195688 号

策划编辑：魏江江
责任编辑：王冰飞
封面设计：刘　键
责任校对：李建庄
责任印制：丛怀宇

出版发行：清华大学出版社
　　网　　址：http://www.tup.com.cn, http://www.wqbook.com
　　地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
　　社 总 机：010-62770175　　**邮　　购：**010-62786544
　　投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
　　质　量　反　馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：北京密云胶印厂
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张：**25.75　　**字　　数：**627 千字
版　　次：2019 年 1 月第 1 版　　**印　　次：**2019 年 1 月第 1 次印刷
印　　数：1～2000
定　　价：79.80 元

产品编号：079718-01

前　言

Python 语言从 20 世纪 90 年代初诞生至今，逐渐被广泛应用于处理系统管理任务和科学计算，是最受欢迎的程序设计语言之一。

学习编程是工程专业学生学习的重要部分，除了直接应用外，学习编程是了解计算机科学本质的方法。计算机科学对现代社会产生了毋庸置疑的影响。Python 是新兴的程序设计语言，是一种解释型、面向对象、动态数据类型的高级程序设计语言。由于 Python 语言简洁、易读并且可扩展，在国外用 Python 做科学计算的研究机构日益增多，最近几年其社会需求逐渐增加，许多国内高校纷纷开设 Python 程序设计课程。本书编者长期从事程序设计语言的教学与应用开发，了解在学习编程的时候什么样的书能够提高 Python 开发能力，以最少的时间投入得到最快的实际应用。

本书内容：

第 1 章是 Python 基础知识，主要讲解 Python 的基础语法和面向对象编程基础、图形界面设计、Python 文件的使用、Python 的第三方库等知识，读者可以轻松掌握。

从第 2 章开始是实用项目案例开发，综合应用前面所学的知识，并且每章都有突出的新知识点，例如侧重数据库应用的案例“智力问答测试”、应用爬虫技术开发的案例“校园网搜索引擎”、应用 itchat 开发的案例“微信机器人”、机器学习案例“基于朴素贝叶斯算法的文本分类”、深度学习案例“基于卷积神经网络的手写体识别”等，还有经典的、大家耳熟能详的游戏案例，例如连连看、推箱子、中国象棋、两人麻将、人物拼图、网络五子棋、飞机大战等。

本书特点：

（1）Python 程序设计涉及的范围非常广泛，本书内容的编排并不求全、求深，而是考虑零基础读者的接受能力，语言的语法介绍以够用、实用为原则，选择 Python 中必备、实用的知识进行讲解，强化对程序思维能力的培养。

（2）案例选取贴近生活，有助于提高读者的学习兴趣。

（3）书中每个案例均提供了详细的设计思路、关键技术分析以及具体的解决方案。

需要说明的是，学习编程是一个实践的过程，而不仅仅是看书、看资料，亲自动手编写、调试程序才是至关重要的。通过实际的编程和积极的思考，读者可以很快地掌握许多宝贵的编程经验，这种编程经验对开发者来说尤其不可或缺。

本书由郑秋生和夏敏捷（中原工学院）主持编写，吴婷（中原工学院）编写第 6 章，韩云飞（中原工学院）编写第 8 章，周延萍编写第 9、10 章，宋宝卫（郑州轻工业学院）编写第 11～15 章，陈海蕊（中原工学院）编写第 16、17 章，高艳霞（中原工学院）编写

第 18 章，李娟（中原工学院）编写第 19 章，郑秋生编写第 20 章，其余章节由夏敏捷编写。在本书的编写过程中，为确保内容的正确性，参阅了很多资料，并且得到了中原工学院的教材资助和资深 Python 程序员的支持，在此谨向他们表示衷心的感谢。本书的学习资源可以扫描封底课件二维码获取。由于编者水平有限，书中难免有不足之处，敬请广大读者批评指正，在此表示感谢。

编　者
2018 年 9 月

目　录

第 1 章

Python 基础知识

Python 是一门跨平台、开源、免费的解释型高级动态编程语言，Python 作为动态语言更适合初学编程者。Python 可以让初学者把精力集中在编程对象和思维方法上，而不用担心语法、类型等外在因素。Python 易于学习，拥有大量的库，可以高效地开发各种应用程序。

1.1 Python 语言简介

Python 由吉多范罗・苏姆（Guido van Rossum）于 1989 年底发明，被广泛应用于处理系统管理任务和科学计算，是最受欢迎的程序设计语言之一。2011 年 1 月，它被 TIOBE 编程语言排行榜评为2010年度语言。自从2004年以后，Python的使用率呈线性增长，TIOBE 最近公布的 2018 年 9 月编程语言指数排行榜中，Python 超越 C++，首次排名处于第三位（前两位是 Java 和 C）。2017 年 7 月，根据 IEEE Spectrum 发布的研究报告显示，Python 已经成为世界上最受欢迎的语言。

Python 支持命令式编程、函数式编程，完全支持面向对象程序设计，语法简洁清晰，并且拥有大量的几乎支持所有领域应用开发的成熟扩展库。

Python 为用户提供了非常完善的基础代码库，覆盖了网络、文件、GUI、数据库、文本等大量内容，用 Python 开发，许多功能不必从零编写，直接使用现成的即可。除了内置的库外，Python 还有大量的第三方库，也就是别人开发的，大家可以直接使用的库。当然，也可以自己开发代码通过很好地封装，作为第三方库给别人使用。Python 就像胶水一样，可以把用多种不同语言编写的程序融合到一起实现无缝拼接，更好地发挥不同语言和工具的优势，满足不同应用领域的需求。所以，Python 程序看上去简单易懂，初学者学 Python，不但入门容易，而且将来深入下去，可以编写那些非常复杂的程序。

Python 同时支持伪编译，将 Python 源程序转换为字节码来优化程序和提高运行速度，可以在没有安装 Python 解释器和相关依赖包的平台上运行。

Python 语言的应用领域主要如下。

（1）Web 开发：Python 语言支持网站开发，比较流行的开发框架有 web2py、Django 等。许多大型网站就是用 Python 开发的，例如 YouTube、Instagram 等。很多大公司，例如 Google、Yahoo 等，甚至 NASA（美国航空航天局）都大量地使用 Python。

（2）网络编程：Python 语言提供了 socket 模块，对 Socket 接口进行了两次封装，支持 Socket 接口的访问；还提供了 urllib、httplib、scrapy 等大量模块，用于对网页内容进行读取和处理，并可以结合多线程编程以及其他有关模块快速开发网页爬虫之类的应用程序；可以使用 Python 语言编写 CGI 程序，也可以把 Python 程序嵌入到网页中运行。

（3）科学计算与数据可视化：Python 中用于科学计算与数据可视化的模块很多，例如 NumPy、SciPy、Matplotlib、Traits、TVTK、Mayavi、VPython、OpenCV 等，涉及的应用领域包括数值计算、符号计算、二维图表、三维数据可视化、三维动画演示、图像处理以及界面设计等。

（4）数据库应用：Python 数据库模块有很多，例如可以通过内置的 sqlite3 模块访问 SQLite 数据库；使用 pywin32 模块访问 Access 数据库；使用 pymysql 模块访问 MySQL 数据库；使用 pywin32 和 pymssql 模块访问 SQL Server 数据库。

视频讲解

（5）多媒体开发：PyMedia 模块可以对 WAV、MP3、AVI 等多媒体格式文件进行编码、解码和播放；PyOpenGL 模块封装了 OpenGL 应用程序编程接口，通过该模块可以在 Python 程序中集成二维或三维图形；PIL（Python Imaging Library，Python 图形库）为 Python 提供了强大的图像处理功能，并提供广泛的图像文件格式支持。

（6）电子游戏应用：Pygame 就是用来开发电子游戏软件的 Python 模块，使用 Pygame 模块可以在 Python 程序中创建功能丰富的游戏和多媒体程序。

Python 有大量的第三方库，可以说需要什么应用就能找到什么 Python 库。

1.2 Python 语法基础

1.2.1 Python 数据类型

视频讲解

计算机程序理所当然地可以处理各种数值。计算机能处理的远不止数值，还可以处理文本、图形、音频、视频、网页等各种各样的数据，不同的数据需要定义不同的数据类型。

❶ 数值类型

Python 数值类型用于存储数值，Python 支持以下数值类型。

- 整型（int）：通常被称为整型或整数，是正或负整数，不带小数点。在 Python3 中只有一种整数类型（int），没有 Python2 中的 long。
- 浮点型（float）：浮点型由整数部分与小数部分组成，浮点型也可以使用科学记数法

表示（2.78E2 就是 2.78×10^2=278）。

- 复数（complex）：复数由实数部分和虚数部分构成，可以用 a+bj 或者 complex(a,b)表示，复数的虚部以字母 j 或 J 结尾，例如 2+3j。

数据类型是不允许改变的，这就意味着如果改变数值数据类型的值，将重新分配内存空间。

❷ 字符串

字符串是 Python 中最常用的数据类型。用户可以使用引号来创建字符串。Python 不支持字符类型，单字符在 Python 也是作为一个字符串使用。Python 使用单引号和双引号来表示字符串是一样的。

❸ 布尔类型

Python 支持布尔类型的数据，布尔类型只有 True 和 False 两种值，但是布尔类型有以下几种运算。

（1）and（与）运算：只有两个布尔值都为 True 时计算结果才为 True。

```
True and True        #结果是 True
True and False       #结果是 False
False and True       #结果是 False
False and False      #结果是 False
```

（2）or（或）运算：只要有一个布尔值为 True，计算结果就是 True。

```
True or True         #结果是 True
True or False        #结果是 True
False or True        #结果是 True
False or False       #结果是 False
```

（3）not（非）运算：把 True 变为 False，或者把 False 变为 True。

```
not True             #结果是 False
not False            #结果是 True
```

布尔运算在计算机中用来做条件判断，根据计算结果为 True 或者 False，计算机可以自动执行不同的后续代码。

在 Python 中，布尔类型还可以与其他数据类型做 and、or 和 not 运算，这时下面的几种情况会被认为是 False：为 0 的数字，包括 0、0.0；空字符串''、""；表示空值的 None；空集合，包括空元组()、空序列[]、空字典{}。其他的值都为 True。例如：

```
a='python'
print(a and True)    #结果是 True
b=''
print(b or False)    #结果是 False
```

❹ 空值

空值是 Python 中的一个特殊值，用 None 表示。它不支持任何运算，也没有任何内置函数方法。None 和任何其他数据类型比较永远返回 False。在 Python 中未指定返回值的函

数会自动返回 None。

1.2.2 序列数据结构

数据结构是计算机存储、组织数据的方式。序列是 Python 中最基本的数据结构。序列中的每个元素都分配一个数字，即它的位置或索引，第 1 个索引是 0，第 2 个索引是 1，依此类推。序列都可以进行的操作包括索引、截取（切片）、加、乘、成员检查。此外，Python 已经内置确定序列的长度以及确定最大和最小元素的方法。Python 内置序列类型最常见的是列表、元组和字符串。另外，Python 提供了字典和集合这样的数据结构，它们属于无顺序的数据集合体，不能通过位置索引来访问数据元素。

❶ 列表

列表（List）是最常用的 Python 数据类型，列表的数据项不需要具有相同的类型。列表类似其他语言的数组，但功能比数组强大得多。

视频讲解

创建一个列表，只要把逗号分隔的不同数据项使用方括号括起来即可，实例如下。

```
list1=['中国', '美国', 1997, 2000];
list2=[1, 2, 3, 4, 5];
list3=["a", "b", "c", "d"];
```

列表索引从 0 开始。列表可以进行截取（切片）、组合等。

1）访问列表中的值

使用下标索引来访问列表中的值，同样可以使用方括号的形式截取字符，实例如下。

```
list1=['中国', '美国', 1997, 2000];
list2=[1, 2, 3, 4, 5, 6, 7];
print("list1[0]: ", list1[0])
print("list2[1:5]: ", list2[1:5])
```

以上实例的输出结果：

```
list1[0]:  中国
list2[1:5]:  [2, 3, 4, 5, 6]
```

2）更新列表

可以对列表的数据项进行修改或更新，实例如下。

```
list=['中国', 'chemistry', 1997, 2000];
print("Value available at index 2 : ")
print(list[2])
list[2]=2001;
print("New value available at index 2 : ")
print(list[2])
```

以上实例的输出结果：

```
Value available at index 2 :
1997
```

```
New value available at index 2 :
2001
```

3）删除列表元素

方法一：使用 del 语句来删除列表中的元素，实例如下。

```
list1=['中国', '美国', 1997, 2000]
print(list1)
del list1[2]
print("After deleting value at index 2 : ")
print(list1)
```

以上实例的输出结果：

```
['中国', '美国', 1997, 2000]
After deleting value at index 2 :
['中国', '美国', 2000]
```

方法二：使用 remove()方法来删除列表中的元素，实例如下。

```
list1=['中国', '美国', 1997, 2000]
list1.remove(1997)
list1.remove('美国')
print(list1)
```

以上实例的输出结果：

```
['中国', 2000]
```

方法三：使用 pop()方法来删除列表中指定位置的元素，无参数时删除最后一个元素，实例如下。

```
list1=['中国', '美国', 1997, 2000]
list1.pop(2)                    #删除位置 2 的元素 1997
list1.pop()                     #删除最后一个元素 2000
print(list1)
```

以上实例的输出结果：

```
['中国', '美国']
```

4）添加列表元素

可以使用 append()方法在列表的末尾添加元素，实例如下。

```
list1=['中国', '美国', 1997, 2000]
list1.append(2003)
print(list1)
```

以上实例的输出结果：

```
['中国', '美国', 1997, 2000, 2003]
```

5）定义多维列表

可以将多维列表视为列表的嵌套，即多维列表的元素值也是一个列表，只是维度比父

列表小一。二维列表（即其他语言的二维数组）的元素值是一维列表，三维列表的元素值是二维列表。例如定义一个二维列表。

```
list2=[["CPU", "内存"], ["硬盘","声卡"]]
```

二维列表比一维列表多一个索引，可以如下获取元素：

```
列表名[索引 1][索引 2]
```

例如定义 3 行 6 列的二维列表，打印出元素值。

```
rows=3
cols=6
matrix=[[0 for col in range(cols)] for row in range(rows)]
                                                #列表生成式生成二维列表
for i in range(rows):
    for j in range(cols):
        matrix[i][j]=i*3+j
        print(matrix[i][j],end=",")
    print('\n')
```

以上实例的输出结果：

```
0,1,2,3,4,5,
3,4,5,6,7,8,
6,7,8,9,10,11,
```

列表生成式（List Comprehensions）是 Python 内置的一种极其强大的生成列表的表达式。如果要生成一个 list [1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9]，可以用 range(1,10)：

```
>>> L=list(range(1,10))            #L 是[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

如果要生成[1×1, 2×2, 3×3 , … , 10×10]，可以使用循环：

```
>>> L=[]
>>> for x in range(1,10):
      L.append(x*x)
>>> L
[1, 4, 9, 16, 25, 36, 49, 64, 81]
```

列表生成式，可以用以下语句代替以上烦琐的循环来完成：

```
>>> [x*x for x in range(1,11)]
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

列表生成式的书写格式为把要生成的元素 x*x 放到前面，后面跟上 for 循环。这样就可以把列表创建出来。for 循环后面还可以加上 if 判断，例如筛选出偶数的平方：

```
>>> [x*x for x in range(1,11) if x%2==0]
[4, 16, 36, 64, 100]
```

再如，把一个列表中的所有字符串变成小写形式：

```
>>> L=['Hello', 'World', 'IBM', 'Apple']
>>> [s.lower() for s in L]
['hello', 'world', 'ibm', 'apple']
```

当然，列表生成式也可以使用两层循环，例如生成'ABC'和'XYZ'中字母的全部组合：

```
>>> print([m+n for m in 'ABC' for n in 'XYZ'])
['AX', 'AY', 'AZ', 'BX', 'BY', 'BZ', 'CX', 'CY', 'CZ']
```

for 循环其实可以同时使用两个甚至多个变量，例如字典（Dict）的 items()可以同时迭代 key 和 value：

```
>>> d={'x': 'A', 'y': 'B', 'z': 'C'}   #字典
>>> for k, v in d.items():
        print(k, '键=', v, endl=';')
```

程序的运行结果：

```
y 键=B; x 键=A; z 键=C;
```

因此，列表生成式也可以使用两个变量来生成列表：

```
>>> d={'x': 'A', 'y': 'B', 'z': 'C'}
>>> [k+'='+v for k, v in d.items()]
['y=B', 'x=A', 'z=C']
```

❷ 元组

Python 的元组（tuple）与列表类似，不同之处在于元组的元素不能修改。元组使用小括号()，列表使用方括号[]。元组中的元素类型也可以不相同。

视频讲解

1）创建元组

元组的创建很简单，只需要在括号中添加元素，并使用逗号隔开即可，实例如下。

```
tup1=('中国', '美国', 1997, 2000)
tup2=(1, 2, 3, 4, 5)
tup3="a", "b", "c", "d"
```

如果是创建空元组，只需写个空括号即可。

```
tup1=()
```

当元组中只包含一个元素时，需要在第 1 个元素后面添加逗号。

```
tup1=(50,)
```

元组与字符串类似，下标索引从 0 开始，可以进行截取、组合等。

2）访问元组

元组可以使用下标索引来访问元组中的值，实例如下。

```
tup1=('中国', '美国', 1997, 2000)
tup2=(1, 2, 3, 4, 5, 6, 7)
```

```
print("tup1[0]: ", tup1[0])              #输出元组的第 1 个元素
print("tup2[1:5]: ", tup2[1:5])          #切片，输出从第 2 个元素开始到第 5 个元素
print(tup2[2:])                          #切片，输出从第 3 个元素开始的所有元素
print(tup2 * 2)                          #输出元组两次
```

以上实例的输出结果：

```
tup1[0]:  中国
tup2[1:5]: (2, 3, 4, 5)
(3, 4, 5, 6, 7)
(1, 2, 3, 4, 5, 6, 7, 1, 2, 3, 4, 5, 6, 7)
```

3）连接元组

元组中的元素值是不允许修改的，但可以对元组进行连接组合，实例如下。

```
tup1=(12, 34, 56)
tup2=(78, 90)
#tup1[0]=100                #修改元组中元素的操作是非法的
tup3=tup1+tup2              #连接元组，创建一个新的元组
print(tup3)
```

以上实例的输出结果：

```
(12, 34, 56, 78, 90)
```

4）删除元组

元组中的元素值是不允许删除的，但可以使用 del 语句来删除整个元组，实例如下。

```
tup=('中国', '美国', 1997, 2000);
print(tup)
del tup
print("After deleting tup : ")
print(tup)
```

以上实例元组被删除后，输出变量会有异常信息，输出如下。

```
('中国', '美国', 1997, 2000)
After deleting tup :
NameError: name 'tup' is not defined
```

5）元组与列表的转换

因为元组数不能改变，所以将元组转换为列表从而可以改变数据。实际上，列表、元组和字符串之间可以互相转换，需要使用 3 个函数，即 str()、tuple()和 list()。

可以使用下面的方法将元组转换为列表：

```
列表对象=list(元组对象)
```

例如：

```
tup=(1, 2, 3, 4, 5)
list1=list(tup)                          #元组转换为列表
print(list1)                             #返回[1, 2, 3, 4, 5]
```

可以使用下面的方法将列表转换为元组：

```
元组对象=tuple(列表对象)
```

例如：

```
nums=[1, 3, 5, 7, 8, 13, 20]
print(tuple(nums))              #列表转换为元组，返回(1, 3, 5, 7, 8, 13, 20)
```

将列表转换成字符串如下：

```
nums=[1, 3, 5, 7, 8, 13, 20]
str1=str(nums)          #列表转换为字符串，返回含中括号及逗号的'[1, 3, 5, 7, 8,
                        #13, 20]'字符串
print(str1[2])      #打印出逗号，因为字符串中索引号为 2 的元素是逗号
num2=['中国', '美国', '日本', '加拿大']
str2="%"
str2=str2.join(num2)    #用百分号连接起来的字符串——'中国%美国%日本%加拿大'
str2=""
str2=str2.join(num2)    #用空字符连接起来的字符串——'中国美国日本加拿大'
```

❸ 字典

Python 字典（dict）是一种可变容器模型，且可存储任意类型对象，例如字符串、数字、元组等其他容器模型。字典也被称为关联数组或哈希表。

视频讲解

1）创建字典

字典由键和对应值（key=>value）成对组成。在字典的每个键/值对中，键和值用冒号分隔，键/值对之间用逗号分隔，整个字典包括在花括号中。其基本语法如下：

```
d={key1: value1, key2: value2}
```

注意：键必须是唯一的，值则不必。值可以取任何数据类型，但键必须是不可变的，例如字符串、数字或元组。

一个简单的字典实例：

```
dict={'xmj' : 40 , 'zhang' : 91 , 'wang' : 80}
```

当然也可以如此创建字典：

```
dict1={'abc': 456};
dict2={'abc': 123, 98.6: 37};
```

字典有如下特性：

（1）字典值可以是任何 Python 对象，例如字符串、数字、元组等。

（2）不允许同一个键出现两次，在创建时如果同一个键被赋值两次，后一个值会覆盖前面的值。

```
dict={'Name': 'xmj', 'Age': 17, 'Name': 'Manni'};
print("dict['Name']: ", dict['Name']);
```

以上实例的输出结果：

```
dict['Name']:  Manni
```

（3）键不可变，所以可以用数字、字符串或元组充当，用列表不行，实例如下。

```
dict={['Name']: 'Zara', 'Age': 7};
```

以上实例输出错误结果：

```
Traceback(most recent call last):
  File "<pyshell#0>", line 1, in <module>
    dict={['Name']: 'Zara', 'Age': 7}
TypeError: unhashable type: 'list'
```

2）访问字典里的值

在访问字典里的值时把相应的键放到方括号中，实例如下。

```
dict={'Name': '王海', 'Age': 17, 'Class': '计算机一班'}
print("dict['Name']: ", dict['Name'])
print("dict['Age']: ", dict['Age'])
```

以上实例的输出结果：

```
dict['Name']:  王海
dict['Age']:  17
```

如果用字典里没有的键访问数据，会输出错误信息：

```
dict={'Name': '王海', 'Age': 17, 'Class': '计算机一班'}
print("dict['sex']: ", dict['sex'])
```

由于没有 sex 键，以上实例输出错误结果：

```
Traceback(most recent call last):
  File "<pyshell#10>", line 1, in <module>
    print("dict['sex']: ", dict['sex'])
KeyError: 'sex'
```

3）修改字典

向字典里添加新内容的方法是增加新的键/值对，修改或删除已有键/值对，实例如下。

```
dict={'Name': '王海', 'Age': 17, 'Class': '计算机一班'}
dict['Age']=18                #更新键/值对
dict['School']="中原工学院"   #增加新的键/值对
print("dict['Age']: ", dict['Age'])
print("dict['School']: ", dict['School'];
```

以上实例的输出结果：

```
dict['Age']:  18
dict['School']:  中原工学院
```

4）删除字典中的元素

del()方法允许使用键从字典中删除元素（条目）。clear()方法清空字典中的所有元素。显式删除一个字典用 del 命令，实例如下。

```
dict={'Name': '王海', 'Age': 17, 'Class': '计算机一班'}
del dict['Name']                #删除键是'Name'的元素（条目）
dict.clear()                    #清空字典中的所有元素
del dict                        #删除字典，用 del 删除后字典不再存在
```

5）in 运算

字典里的 in 运算用于判断某键是否在字典里，对于 value 值不适用，其功能与 has_key (key)相似。

```
dict={'Name': '王海', 'Age': 17, 'Class': '计算机一班'}
print('Age' in dict)          #等价于 print(dict.has_key('Age'))
```

以上实例的输出结果：

```
True
```

6）获取字典中的所有值

dict.values()以列表形式返回字典中的所有值。

```
dict={'Name': '王海', 'Age': 17, 'Class': '计算机一班'}
print(dict.values())
```

以上实例的输出结果：

```
[17, '王海', '计算机一班']
```

7）items()方法

items()方法把字典中的每对 key 和 value 组成一个元组，并把这些元组放在列表中返回。

```
dict={'Name': '王海', 'Age': 17, 'Class': '计算机一班'}
for key,value in dict.items():
    print(key,value)
```

以上实例的输出结果：

```
Name 王海
Class 计算机一班
Age 17
```

注意：字典打印出来的顺序与创建之初的顺序不同，这不是错误。字典中的各个元素并没有顺序之分（因为不需要通过位置查找元素），所以在存储元素时进行了优化，使字典的存储和查询效率最高。这也是字典和列表的另一个区别：列表保持元素的相对关系，即序列关系；而字典是完全无序的，也称为非序列。如果想保持一个集合中元素的顺序，需要使用列表，而不是字典。

❹ 集合

集合（set）是一个无序不重复元素的序列。集合的基本功能是进行成员关系测试和删

除重复元素。

1）创建集合

可以使用大括号（{}）或者 set()函数创建集合。注意，创建一个空集合必须用 set()，而不是用{}，因为{}用来创建一个空字典。

```
student={'Tom', 'Jim', 'Mary', 'Tom', 'Jack', 'Rose'}
print(student)   #输出集合，重复的元素被自动去掉
```

以上实例的输出结果：

```
{'Jack', 'Rose', 'Mary', 'Jim', 'Tom'}
```

2）成员测试

例如：

```
if('Rose' in student) :
    print('Rose 在集合中')
else :
    print('Rose 不在集合中')
```

以上实例的输出结果：

```
Rose 在集合中
```

3）集合运算

可以使用"–""|""&"运算符进行集合的差集、并集、交集运算，例如：

```
#set 可以进行集合运算
a=set('abcd')
b=set('cdef')
print(a)
print("a 和 b 的差集：", a - b)                #a 和 b 的差集
print("a 和 b 的并集：", a | b)                #a 和 b 的并集
print("a 和 b 的交集：", a & b)                #a 和 b 的交集
print("a 和 b 中不同时存在的元素：", a ^ b)      #a 和 b 中不同时存在的元素
```

以上实例的输出结果：

```
{'a', 'c', 'd', 'b'}
a 和 b 的差集：{'a', 'b'}
a 和 b 的并集：{'b', 'a', 'f', 'd', 'c', 'e'}
a 和 b 的交集：{'c', 'd'}
a 和 b 中不同时存在的元素：{'a', 'e', 'f', 'b'}
```

1.2.3 Python 控制语句

对于 Python 程序中的执行语句，默认是按照书写顺序依次执行的，通常说这样的语句是顺序结构的。但是，仅有顺序结构还不够，因为有时候需要根据特定的情况有选择地执

行某些语句，这时就需要一种选择结构的语句。另外，有时候还可以在给定条件下重复执行某些语句，通常称这些语句是循环结构的。有了这 3 种基本结构，就能构建任意复杂的程序了。

视频讲解

❶ 选择结构

3 种基本程序结构中的选择结构，可以用 if 语句、if…else 语句和 if…elif…else 语句实现。

if 语句是一种单选结构，它选择的是做与不做。if 语句的语法形式如下：

```
if 表达式:
    语句 1
```

if 语句的流程图如图 1-1 所示。

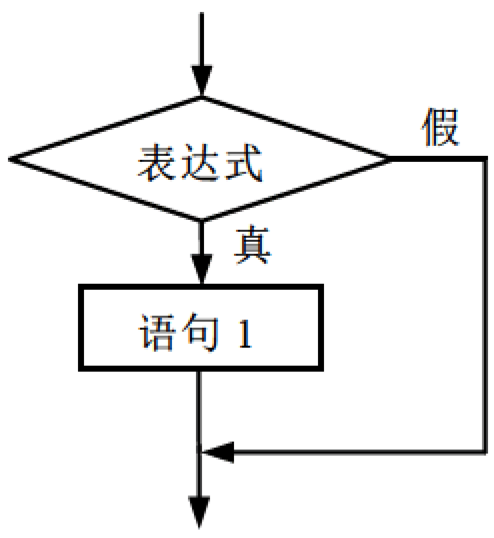


图 1-1　if 语句的流程图

if…else 语句是一种双选结构，用于解决在两种备选行动中选择哪一个的问题。if…else 语句的语法形式如下：

```
if 表达式:
    语句 1
else:
    语句 2
```

if…else 语句的流程图如图 1-2 所示。

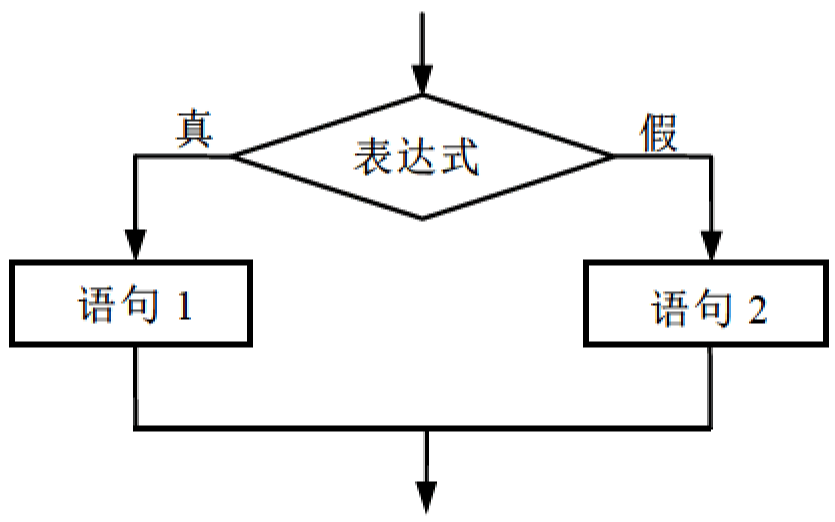


图 1-2　if…else 语句的流程图

例 1-1　输入一个年份，判断是否为闰年。

闰年的年份必须满足以下两个条件之一：

（1）能被 4 整除，但不能被 100 整除。

（2）能被 400 整除。

分析：设变量 year 表示年份，判断 year 是否满足以下表达式。

条件（1）的逻辑表达式是 year%4==0&&year%100 !=0。

条件（2）的逻辑表达式是 year%400==0。

两者取“或”，即得到判断闰年的逻辑表达式：

```
(year%4==0 and year%100!=0)  or  year%400==0
```

程序代码：

```
year=int(input('输入年份:')) #输入 x，input()获取的是字符串，所以需要转换成整型
if  year%4==0 and year%100!=0 or year%400==0:  #注意运算符的优先级
    print(year, "是闰年")
else:
    print(year, "不是闰年")
```

在判断闰年后，也可以输入某年某月某日，判断这一天是这一年的第几天。以 3 月 5 日为例，应该先把前两个月的天数加起来，然后再加上 5 天，即本年的第几天。特殊情况是闰年，在输入月份大于 3 时需考虑多加一天。

程序代码：

```
year=int(input('year:'))                     #输入年
month=int(input('month:'))                   #输入月
day=int(input('day:'))                       #输入日
months=(0,31,59,90,120,151,181,212,243,273,304,334)
if 0<=month<=12:
    sum=months[month - 1]
else:
    print('月份输入错误')
sum+=day
leap=0
if(year%400==0) or ((year%4==0) and(year%100!=0)):
    leap=1
if(leap==1) and(month>2):
    sum+=1
print('这一天是这一年的第%d 天'%sum)
```

有时候需要在多组动作中选择一组执行，这时就要用到多选结构，对于 Python 语言来说就是 if…elif…else 语句。该语句的语法形式如下：

```
if 表达式 1:
    语句 1
elif 表达式 2:
    语句 2
    …
elif 表达式 n:
    语句 n
else:
```

```
    语句 n+1
```

注意：最后一个 elif 子句之后的 else 子句没有进行条件判断，它实际上处理跟前面所有条件都不匹配的情况，所以 else 子句必须放在最后。if…elif…else 语句的流程图如图 1-3 所示。

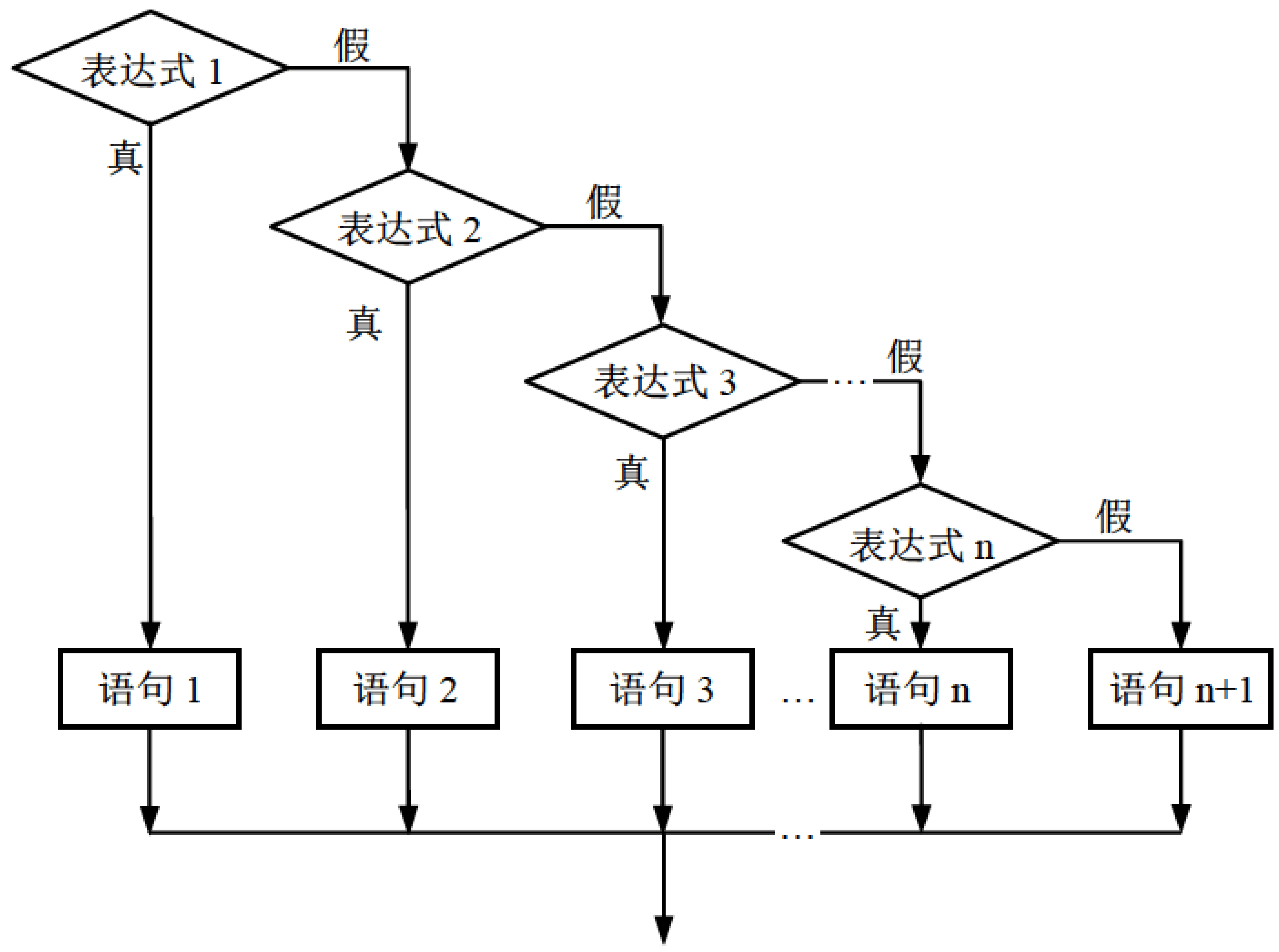


图 1-3　if…elif…else 语句的流程图

例 1-2　输入学生的成绩 score，按分数输出其等级，即 score≥90 为优，90>score≥80 为良，80>score≥70 为中等，70>score≥60 为及格，score<60 为不及格。

```
score=int(input("请输入成绩"))      #int()转换字符串为整型
if score>=90:
   print("优")
elif  score>=80:
   print("良")
elif  score>=70:
   print("中")
elif  score>=60:
   print("及格")
else :
   print("不及格")
```

说明：在 3 种选择语句中，条件表达式都是必不可少的组成部分。当条件表达式的值为零时，表示条件为假；当条件表达式的值为非零时，表示条件为真。那么哪些表达式可以作为条件表达式呢？最常用的是关系表达式和逻辑表达式，例如：

```
if  a==x  and  b==y :
   print("a=x, b=y")
```

除此之外，条件表达式可以是任何数值类型的表达式，字符串也可以：

```
if  'a':   #'abc':也可以
    print("a=x, b=y")
```

另外，C 语言用花括号{}来区分语句体，但 Python 的语句体是用缩进形式来表示的，如果缩进不正确，会导致逻辑错误。

❷ 循环结构

程序在一般情况下是按顺序执行的。编程语言提供了各种控制结构，允许更复杂的执行路径。循环语句允许用户执行一个语句或语句组多次，Python 提供了 for 循环和 while 循环（在 Python 中没有 do…while 循环）。

视频讲解

1）while 语句

在 Python 编程中，while 语句用于循环执行程序，即在某条件下循环执行某段程序，以处理需要重复处理的相同任务。其基本形式如下：

```
while 判断条件:
    执行语句
```

执行语句可以是单个语句或语句块。判断条件可以是任何表达式，任何非零或非空的值均为 True。当判断条件为 False 时，循环结束。while 语句的流程图如图 1-4 所示。

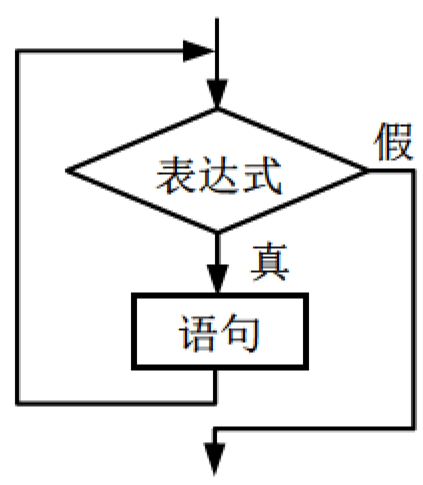


图 1-4　while 语句的流程图

同样需要注意冒号和缩进，例如：

```
count=0
while count<5:
   print('The count is:', count)
   count=count+1
print("Good bye!")
```

2）for 语句

for 语句可以遍历任何序列的项目，例如一个列表、元组或者一个字符串。for 循环的语法格式如下：

视频讲解

```
for 循环索引值 in 序列
    循环体
```

for 循环会把列表中的元素遍历出来，例如以下代码会依次打印 fruits 中的每一个元素。

```
fruits=['banana', 'apple', 'mango']
for fruit in fruits:          #第 2 个实例
```

```
    print('元素:', fruit)
print("Good bye!")
```

程序的运行结果：

```
元素：banana
元素：apple
元素：mango
Good bye!
```

例1-3 计算 1～10 的整数之和，可以用一个 sum 变量做累加。

程序代码：

```
sum=0
for x in [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]:
    sum=sum+x
print(sum)
```

如果要计算 1～100 的整数之和，从 1 写到 100 有点困难。Python 提供了 range()内置函数，可以生成一个整数序列，再通过 list()函数转换成列表。

例如，range(0, 5)或 range(5)生成的序列是从 0 开始小于 5 的整数，不包括 5。

```
>>> list(range(5))
[0, 1, 2, 3, 4]
```

range(1, 101)就可以生成 1～100 的整数序列，计算 1～100 的整数之和如下。

```
sum=0
for x in range(1,101):
    sum=sum+x
print(sum)
```

3）continue 和 break 语句

break 语句在 while 循环和 for 循环中都可以使用，一般放在 if 选择结构中，一旦 break 语句被执行，将使整个循环提前结束。

continue 语句的作用是终止当前循环，并忽略 continue 之后的语句，然后回到循环的顶端，提前进入下一次循环。

除非 break 语句能让代码更简单或更清晰，否则不要轻易使用。

例1-4 continue 和 break 用法示例。

```
#continue 和 break 用法
i=1
while i<10:
    i+=1
    if i%2>0:               #非双数时跳过输出
        continue
    print(i)                #输出双数 2、4、6、8、10

i=1
while 1:                    #循环条件为 1 必定成立
```

```
    print(i)                  #输出 1~10
    i+=1
    if i>10:                  #当 i 大于 10 时跳出循环
        break
```

在 Python 程序开发的过程中，将完成某一特定功能并经常使用的代码编写成函数，放在函数库（模块）中供大家选用，在需要使用时直接调用，这就是程序中的函数。开发人员要善于使用函数，以提高编码效率，减少编写程序段的工作量。

1.2.4 Python 函数与模块

当某些任务（例如求一个数的阶乘）需要在一个程序中的不同位置重复执行时，会造成代码的重复率高，应用程序代码烦琐。解决这个问题的方法就是使用函数。无论是在哪门编程语言中，函数（在类中称为方法，意义是相同的）都扮演着至关重要的角色。模块是 Python 的代码组织单元，它将函数、类和数据封装起来以便重用，模块往往对应 Python 程序文件，Python 标准库和第三方提供了大量的模块。

❶ 函数的定义

在 Python 中，函数定义的基本形式如下：

```
def  函数名(函数参数):
    函数体
    return 表达式或者值
```

在这里说明几点：

（1）在 Python 中采用 def 关键字进行函数的定义，不用指定返回值的类型。

（2）函数参数可以是零个、一个或者多个，同样，函数参数也不用指定参数类型，因为在 Python 中变量都是弱类型的，Python 会自动根据值来维护其类型。

（3）在 Python 中，函数定义中的缩进部分是函数体。

（4）函数的返回值是通过函数中的 return 语句获得的。return 语句是可选的，它可以在函数体内的任何地方出现，表示函数调用执行到此结束；如果没有 return 语句，会自动返回 None（空值），如果有 return 语句，但是 return 后面没有接表达式或者值，也是返回 None（空值）。

下面定义 3 个函数：

```
def printHello():             #打印'hello'字符串
    print('hello')

def printNum():               #输出数字 0~9
    for i in range(0,10):
        print(i)
    return

def add(a,b):                 #实现两个数的和
```

视频讲解

```
    return a+b
```

❷ 函数的使用

在定义了函数之后就可以使用该函数了，但是在 Python 中要注意一个问题，就是在 Python 中不允许前向引用，即在函数定义之前不允许调用该函数。大家看下面的例子就明白了：

```
print(add(1,2))
def add(a,b):
    return a+b
```

视频讲解

这段程序运行的错误提示如下：

```
Traceback(most recent call last):
  File "C:/Users/xmj/4-1.py", line 1, in <module>
    print(add(1,2))
NameError: name 'add' is not defined
```

从报的错可以知道，名字为 add 的函数未进行定义。所以在任何时候调用某个函数，必须确保其定义在调用之前。

例 1-5　编写函数，计算形如 a+aa+aaa+aaaa +…+ aaa…aaa 的表达式的值，其中 a 为小于 10 的自然数。例如 2+22+222+2222+22222（此时 n=5），a、n 由用户从键盘输入。

分析：关键是计算出求和中每一项的值。容易看出每一项都是前一项扩大 10 倍后加 a。

程序代码：

```
def  sum(a, n):
    result, t=0, 0        #同时将 result、t 赋值为 0，这种形式比较简洁
    for i in range(n):
        t=t*10+a
        result+=t
    return result
#用户输入两个数字
a=int(input("输入 a: "))
n=int(input("输入 n: "))
print(sum(a, n))
```

程序运行结果：

```
输入 a: 2↙
输入 n: 5↙
24690
```

❸ 闭包

在 Python 中，闭包（closure）指函数的嵌套。用户可以在函数内部定义一个嵌套函数，将嵌套函数视为一个对象，所以可以将嵌套函数作为定义它的函数的返回结果。

视频讲解

例 1-6　使用闭包的例子。

```
def func_lib():
    def add(x, y):
        return x+y
    return add       #返回函数对象

fadd=func_lib()
print(fadd(1, 2))
```

在函数 func_lib()中定义了一个嵌套函数 add(x, y)，并作为函数 func_lib()的返回值。其运行结果为 3。

❹ 函数的递归调用

函数在执行的过程中直接或间接调用自己本身，称为递归调用。Python 语言允许递归调用。

例 1-7 求 1～5 的平方和。

```
def f(x):
  if x==1:                       #递归调用结束的条件
    return 1
  else:
    return(f(x-1)+x*x)           #调用 f()函数本身
print(f(5))
```

❺ 模块

通过模块（module）能够有逻辑地组织 Python 代码段，把相关的代码分配到一个模块里能让代码更好用、更易懂。简单地说，模块就是一个保存了 Python 代码的文件。在模块里能定义函数、类和变量。

视频讲解

在 Python 中，模块和 C 语言中的头文件以及 Java 中的包很类似，例如在 Python 中要调用 sqrt()函数，必须用 import 关键字引入 math 这个模块。

1）导入某个模块

在 Python 中用关键字 import 来导入某个模块，方式如下：

```
import 模块名            #导入模块
```

例如要引用模块 math，就可以在文件最开始的地方用 import math 来导入。

在调用模块中的函数时必须这样调用：

```
模块名.函数名
```

例如：

```
import math              #导入 math 模块
print("50 的平方根：", math.sqrt(50))
```

为什么必须加上模块名这样调用呢？因为可能存在这样一种情况：在多个模块中含有相同名称的函数，此时如果只是通过函数名来调用，解释器无法知道到底要调用哪个函数。所以如果像上述这样导入模块，调用函数必须加上模块名。

有时候只需要用到模块中的某个函数，此时只要引入该函数即可，通过 from 语句实现：

```
from 模块名 import 函数名1,函数名2…
```

通过这种方式引入，在调用函数时只能给出函数名，不能给出模块名，但是当两个模块中含有相同名称函数的时候，后面一次引入会覆盖前一次引入。

也就是说，假如模块 A 中有函数 fun()，在模块 B 中也有函数 fun()，如果引入 A 中的 fun()在先，B 中的 fun()在后，那么在调用 fun()函数的时候会去执行模块 B 中的 fun()函数。

如果想一次性导入 math 中的所有东西，还可以通过以下语句实现：

```
from math import *
```

这是一种简单的导入模块中所有项目的方式，然而不建议过多地使用这种方式。

2）定义自己的模块

在 Python 中，每个 Python 文件都可以作为一个模块，模块的名字就是文件的名字。

比如有这样一个文件 fibo.py，在 fibo.py 中定义了 3 个函数 add()、fib()、fib2()：

```
#fibo.py
#斐波那契（Fibonacci）数列模块
def fib(n):          #定义到 n 的斐波那契数列
    a, b=0, 1
    while b<n:
        print(b, end=' ')
        a, b=b, a+b
    print()
def fib2(n):         #返回到 n 的斐波那契数列
    result=[]
    a, b=0, 1
    while b<n:
        result.append(b)
        a, b=b, a+b
    return result
def add(a,b):
    return a+b
```

那么在其他文件（例如 test.py）中就可以如下使用：

```
#test.py
import fibo
```

加上模块名称来调用函数：

```
fibo.fib(1000)       #结果是 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
fibo.fib2(100)       #结果是[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
test.add(2,3)        #结果是 5
```

当然，也可以通过 from fibo import add, fib, fib2 来引入。

直接通过函数名来调用函数：

```
fib(500)          #结果是 1  1  2  3  5  8  13  21  34  55  89  144  233  377
```

如果想列举 fibo 模块中定义的属性，代码如下：

```
import fibo
dir(fibo)          #得到自定义模块 fibo 中定义的变量和函数
```

输出结果：

```
['__name__', 'fib', 'fib2', 'add']
```

1.3　Python 面向对象设计

面向对象程序设计（Object Oriented Programming，OOP）的思想主要针对大型软件设计而提出，使得软件设计更加灵活，能够很好地支持代码复用和设计复用，并且使得代码具有更好的可读性和可扩展性。

现实生活中的每一个相对独立的事物都可以看作一个对象，例如一个人、一辆车、一台计算机等。对象是具有某些特性和功能的具体事物的抽象。每个对象都具有描述其特征的属性及附属于它的行为。例如，一辆车有颜色、车轮数、座椅数等属性，也有启动、行驶、停止等行为；一个人由姓名、性别、年龄、身高、体重等特征描述，也有走路、说话、学习、开车等行为；一台计算机由主机、显示器、键盘、鼠标等部件组成。

在人们生产一台计算机的时候，并不是先生产主机再生产显示器再生产键盘和鼠标，即不是顺序执行的，而是分别生产设计主机、显示器、键盘、鼠标等，最后把它们组装起来。这些部件通过事先设计好的接口连接，以便协调地工作。这就是面向对象程序设计的基本思路。

每个对象都有一个类型，类是创建对象实例的模板，是对对象的抽象和概括，它包含对所创建对象的属性描述和行为特征的定义。例如，马路上的汽车是一个一个的汽车对象，它们归属于一个汽车类，那么车身颜色就是该类的属性，开动是它的方法，该保养了或者该报废了就是它的事件。

Python 完全采用了面向对象程序设计的思想，是真正面向对象的高级动态编程语言，完全支持面向对象的基本功能，例如封装、继承、多态以及对基类方法的覆盖或重写。与其他面向对象程序设计语言不同的是，Python 中对象的概念很广泛，Python 中的一切内容都可以称为对象。例如，字符串、列表、字典、元组等内置数据类型都具有和类完全相似的语法和用法。

1.3.1　定义和使用类

视频讲解

❶ 类的定义

在创建类时用变量形式表示的对象属性称为数据成员或属性（成员变量），用函数形式表示的对象行为称为成员函数（成员方法），成员属性和成员方法统称为类的成员。

类定义的最简单的形式如下：

```
class 类名:
    属性（成员变量）
    属性
    …
    成员函数（成员方法）
```

例如定义一个 Person 人员类：

```
class Person:
    num=1                        #成员变量（属性）
    def SayHello(self):          #成员函数
        print("Hello!");
```

这里在 Person 类中定义了一个成员函数 SayHello(self)，用于输出字符串"Hello!"。同样，Python 使用缩进标识类的定义代码。

❷ **对象的创建**

对象是类的实例。如果人类是一个类，那么某个具体的人就是一个对象。只有定义了具体的对象，才能通过“对象名.成员”的方式来访问其中的数据成员或成员方法。

Python 创建对象的语法如下：

```
对象名=类名()
```

例如，下面的代码定义了一个 Person 类的对象 p：

```
p=Person()
p.SayHello()                #访问成员函数 SayHello()
```

运行结果如下：

```
Hello!
```

1.3.2　构造函数

类可以定义一个特殊的称为__init__()的方法（构造函数，以两个下画线“__”开头和结束）。在一个类定义了__init__()方法以后，类实例化时就会自动为新生成的类实例调用__init__()方法。构造函数一般用于完成对象数据成员设置初值或进行其他必要的初始化工作。如果用户未涉及构造函数，Python 将提供一个默认的构造函数。

例如定义一个复数类 Complex，构造函数完成对象变量的初始化工作。

```
class Complex:
    def __init__(self, realpart, imagpart):
        self.r=realpart
        self.i=imagpart
x=Complex(3.0,-4.5)
print(x.r, x.i)
```

运行结果如下：

```
3.0  -4.5
```

1.3.3　析构函数

Python 中类的析构函数是_ _del_ _()，用来释放对象占用的资源，在 Python 收回对象空间之前自动执行。如果用户未涉及析构函数，Python 将提供一个默认的析构函数进行必要的清理工作。

例如：

```
class Complex:
    def __init__(self, realpart, imagpart):
        self.r=realpart
        self.i=imagpart
    def  __del__(self):
        print("Complex 不存在了")
x=Complex(3.0,-4.5)
print(x.r, x.i)
print(x)
del  x                              #删除对象变量 x
```

运行结果如下：

```
3.0 -4.5
<__main__.Complex object at 0x01F87C90>
Complex 不存在了
```

说明：在删除对象变量 x 之前，x 是存在的，在内存中的标识为 0x01F87C90，执行“del x”语句后，对象变量 x 不存在了，系统自动调用析构函数，所以出现“Complex 不存在了”的情况。

1.3.4　实例属性和类属性

属性（成员变量）有两种，一种是实例属性，另一种是类属性（类变量）。实例属性是在构造函数_ _init_ _()（以两个下画线“_ _”开头和结束）中定义的，定义时以 self 作为前缀；类属性是在类中方法之外定义的属性。在主程序中（在类的外部），实例属性属于实例（对象）。只能通过对象名访问；类属性属于类，可以通过类名访问，也可以通过对象名访问，为类的所有实例共享。

例 1-8　定义含有实例属性（姓名 name、年龄 age）和类属性（人数 num）的 Person（人员）类。

```
class Person:
     num=1                              #类属性
     def __init__(self, str,n):         #构造函数
         self.name=str                  #实例属性
```

```
        self.age=n
    def SayHello(self):              #成员函数
        print("Hello!")
    def PrintName(self):             #成员函数
        print("姓名: ", self.name, "年龄: ", self.age)
    def PrintNum(self):              #成员函数
        print(Person.num)            #由于是类属性，所以不写 self.num
#主程序
P1=Person("夏敏捷",42)
P2=Person("王琳",36)
P1.PrintName()
P2.PrintName()
Person.num=2                         #修改类属性
P1.PrintNum()
P2.PrintNum()
```

运行结果如下：

```
姓名:  夏敏捷 年龄:  42
姓名:  王琳 年龄:  36
2
2
```

num 变量是一个类变量，它的值将在这个类的所有实例之间共享。用户可以在类内部或类外部使用 Person. num 访问。

在类的成员函数（方法）中可以调用类的其他成员函数（方法），可以访问类属性、对象实例属性。

在 Python 中比较特殊的是，可以动态地为类和对象增加成员，这一点是和很多面向对象程序设计语言不同的，也是 Python 动态类型特点的一种重要体现。

1.3.5　私有成员与公有成员

Python 并没有对私有成员提供严格的访问保护机制。在定义类的属性时，如果属性名以两个下画线“_ _”开头，表示是私有属性，否则是公有属性。私有属性在类的外部不能直接访问，需要通过调用对象的公有成员方法来访问，或者通过 Python 支持的特殊方式来访问。Python 提供了访问私有属性的特殊方式，可用于程序的测试和调试，对于成员方法也具有同样的性质。这种方式如下：

```
对象名._类名+私有成员
```

例如访问 Car 类私有成员_ _weight：

```
car1._Car__weight
```

私有属性是为了数据封装和保密而设的属性，一般只能在类的成员方法（类的内部）中使用访问，虽然 Python 支持一种特殊的方式从外部直接访问类的私有成员，但是并不推

荐大家这样做。公有属性是可以公开使用的，既可以在类的内部进行访问，也可以在外部程序中使用。

例 1-9 为 Car 类定义私有成员。

```
class Car:
    price=100000                      #定义类属性
    def __init__(self, c, w):
        self.color=c                  #定义公有属性 color
        self. __weight=w              #定义私有属性__weight
#主程序
car1=Car("Red",10.5)
car2=Car("Blue",11.8)
print(car1.color)
print(car1. __Car__weight)
print(car1. __weight)                 #AttributeError
```

运行结果如下：

```
Red
10.5
AttributeError: 'Car' object has no attribute '__weight'
```

1.3.6 方法

在类中定义的方法可以粗略地分为 3 大类，即公有方法、私有方法、静态方法。其中，公有方法、私有方法都属于对象，私有方法的名字以两个下画线“__”开始，每个对象都有自己的公有方法和私有方法，在这两类方法中可以访问属于类和对象的成员；公有方法通过对象名直接调用，私有方法不能通过对象名直接调用，只能在属于对象的方法中通过 self 调用或在外部通过 Python 支持的特殊方式来调用。如果通过类名来调用属于对象的公有方法，需要显式地为该方法的 self 参数传递一个对象名，用来明确指定访问哪个对象的数据成员。静态方法可以通过类名和对象名调用，但不能直接访问属于对象的成员，只能访问属于类的成员。

例 1-10 公有方法、私有方法、静态方法的定义和调用。

```
class Fruit:
    price=0
    def __init__(self):
        self.__color='Red'            #定义和设置私有属性 color
        self.__city='Kunming'         #定义和设置私有属性 city
    def __outputColor(self):          #定义私有方法 outputColor()
        print(self.__color)           #访问私有属性 color
    def __outputCity(self):           #定义私有方法 outputCity()
       print(self.__city)             #访问私有属性 city
    def output(self):                 #定义公有方法 output()
        self.__outputColor()          #调用私有方法 outputColor()
```

```
            self.__outputCity()                    #调用私有方法 outputCity()
        @ staticmethod
        def getPrice():                            #定义静态方法 getPrice()
             return Fruit.price
        @ staticmethod
        def setPrice(p):                           #定义静态方法 setPrice()
            Fruit.price=p
    #主程序
    apple=Fruit()
    apple.output()
    print(Fruit.getPrice())
    Fruit.setPrice(9)
    print(Fruit.getPrice())
```

运行结果如下：

```
Red
Kunming
0
9
```

继承是为代码复用和设计复用而设计的，是面向对象程序设计的重要特性之一。在设计一个新类时，如果可以继承一个已有的设计良好的类，然后进行二次开发，无疑会大幅度减少开发的工作量。

1.3.7　类的继承

在继承关系中，已有的、设计好的类称为父类或基类，新设计的类称为子类或派生类。派生类可以继承父类的公有成员，但是不能继承其私有成员。

视频讲解

类继承的语法格式如下：

```
class 派生类名(基类名):                #基类名写在括号里
    派生类成员
```

在 Python 中继承有以下特点：

（1）在继承中基类的构造函数（__init__()方法）不会被自动调用，它需要在其派生类的构造中专门调用。

（2）如果需要在派生类中调用基类的方法，通过“基类名.方法名()”的方式来实现，需要加上基类的类名前缀，且需要带上 self 参数变量（在类中调用普通函数时并不需要带上 self 参数），也可以使用内置函数 super()实现这一目的。

（3）Python 总是先查找对应类型的方法，如果不能在派生类中找到对应的方法，它才开始到基类中逐个查找（先在本类中查找调用的方法，找不到才去基类中找）。

例 **1-11**　设计 Person 类，并根据 Person 派生 Student 类，分别创建 Person 类与 Student 类的对象。

```
#定义基类：Person 类
import types
class Person(object):#基类必须继承于 object，否则在派生类中将无法使用 super()函数
    def __init__(self, name='', age=20, sex='man'):
        self.setName(name)
        self.setAge(age)
        self.setSex(sex)
    def setName(self, name):
        if type(name)!=str:      #内置函数 type()返回被测对象的数据类型
            print('姓名必须是字符串.')
            return
        self.__name=name
    def setAge(self, age):
        if type(age)!=int:
            print('年龄必须是整型.')
            return
        self.__age=age
    def setSex(self, sex):
        if sex!='男' and sex!='女':
            print('性别输入错误')
            return
        self.__sex=sex
    def show(self):
        print('姓名：', self.__name, '年龄：', self.__age ,'性别：', self.__sex)
#定义子类（Student 类），在其中增加一个入学年份私有属性（数据成员）
class Student(Person):
    def __init__(self, name='', age=20, sex='man', schoolyear=2016):
        #调用基类构造方法初始化基类的私有数据成员
        super(Student, self).__init__(name, age, sex)
        #Person.__init__(self, name, age, sex)#也可以这样初始化基类的私有数据成员
        self.setSchoolyear(schoolyear)      #初始化派生类的数据成员
    def setSchoolyear(self, schoolyear):
        self.__schoolyear=schoolyear
    def show(self):
        Person.show(self)                   #调用基类的 show()方法
        #super(Student, self).show()        #也可以这样调用基类的 show()方法
        print('入学年份：', self.__schoolyear)
#主程序
if __name__=='__main__':
    zhangsan=Person('张三', 19, '男')
    zhangsan.show()
    lisi=Student('李四', 18, '男', 2015)
    lisi.show()
    lisi.setAge(20)                         #调用继承的方法修改年龄
    lisi.show()
```

运行结果如下：

```
姓名： 张三 年龄： 19 性别： 男
姓名： 李四 年龄： 18 性别： 男
入学年份：2015
姓名： 李四 年龄： 20 性别： 男
入学年份：2015
```

方法重写必须出现在继承中。它是指在派生类继承了基类的方法之后，如果基类方法的功能不能满足需求，需要对基类中的某些方法进行修改。用户可以在派生类中重写基类的方法，这就是重写。

例 1-12　重写父类（基类）的方法。

```
class Animal:                          #定义父类
   def run(self):
      print(Animal is running...)      #调用父类方法
class Cat(Animal):                     #定义子类
   def run(self):
      print(Cat is running...)         #调用子类方法
class Dog(Animal):                     #定义子类
   def run(self):
      print(Dog is running...)         #调用子类方法

c=Dog()                                #子类实例
c.run()                                #子类调用重写方法
```

程序运行结果：

```
Dog is running...
```

当子类 Dog 和父类 Animal 存在相同的 run()方法时，子类的 run()覆盖了父类的 run()，在代码运行时总是会调用子类的 run()，这样就获得了继承的另一个优点——多态。

1.3.8　多态

视频讲解

要理解什么是多态，首先要对数据类型再做一点说明：在定义一个 class 的时候，实际上就定义了一种数据类型。通常，定义的数据类型和 Python 自带的数据类型（例如 string、list、dict）没什么区别。

```
a=list()               #a 是 list 类型
b=Animal()             #b 是 Animal 类型
c=Dog()                #c 是 Dog 类型
```

一个变量是否为某个类型可以用 isinstance()判断：

```
>>> isinstance(a, list)
True
```

```
>>> isinstance(b, Animal)
True
>>> isinstance(c, Dog)
True
```

a、b、c 确实对应 list、Animal、Dog 这 3 种类型。

```
>>> isinstance(c, Animal)
True
```

因为 Dog 是从 Animal 继承的，当创建了一个 Dog 的实例 c 时，认为 c 的数据类型是 Dog 没错，但 c 同时也是 Animal，Dog 本来就是 Animal 的一种。

所以，在继承关系中，如果一个实例的数据类型是某个子类，那么它的数据类型也可以被看作是父类。但是，反过来就不可以：

```
>>> b=Animal()
>>> isinstance(b, Dog)
False
```

Dog 可以看成 Animal，但 Animal 不可以看成 Dog。

要理解多态的好处，还需要再编写一个函数，这个函数接受一个 Animal 类型的变量：

```
def run_twice(animal):
    animal.run()
    animal.run()
```

当传入 Animal 的实例时，run_twice()就打印出：

```
>>> run_twice(Animal())
Animal is running...
Animal is running...
```

当传入 Dog 的实例时，run_twice()就打印出：

```
>>> run_twice(Dog())
Dog is running...
Dog is running...
```

当传入 Cat 的实例时，run_twice()就打印出：

```
>>> run_twice(Cat())
Cat is running...
Cat is running...
```

现在，如果再定义一个 Tortoise 类型，也从 Animal 派生：

```
class Tortoise(Animal):
    def run(self):
        print('Tortoise is running slowly...')
```

当调用 run_twice()时传入 Tortoise 的实例：

```
>>> run_twice(Tortoise())
```

```
Tortoise is running slowly...
Tortoise is running slowly...
```

大家会发现新增了一个 Animal 的子类，不必对 run_twice()做任何修改。实际上，任何依赖 Animal 作为参数的函数或者方法都可以不加修改地正常运行，原因就在于多态。

多态的好处就是，当需要传入 Dog、Cat、Tortoise 等时，只需要接收 Animal 类型就可以了，因为 Dog、Cat、Tortoise 等都是 Animal 类型，然后按照 Animal 类型进行操作。由于 Animal 类型有 run()方法，因此传入的任意类型，只要是 Animal 类或者子类，就会自动调用实际类型的 run()方法，这就是多态的意思。

对于一个变量，用户只需要知道它是 Animal 类型，无须确切地知道它的子类型，就可以放心地调用 run()方法，而具体调用的 run()方法是作用在 Animal、Dog、Cat 还是 Tortoise 对象上，由运行时该对象的确切类型决定，这就是多态真正的威力：调用方只管调用，不管细节，而当新增一种 Animal 的子类时，只要确保 run()方法编写正确，不用管原来的代码是如何调用的。这就是著名的“开闭”原则，包括以下两点。

（1）对扩展开放：允许新增 Animal 子类。

（2）对修改封闭：不需要修改依赖 Animal 类型的 run_twice()等函数。

1.3.9 面向对象应用案例——扑克牌发牌程序

【案例】采用扑克牌类设计扑克牌发牌程序。

4 名牌手打牌，计算机随机将 52 张牌（不含大/小鬼）发给 4 名牌手，并在屏幕上显示每位牌手的牌。程序的运行效果如图 1-5 所示。

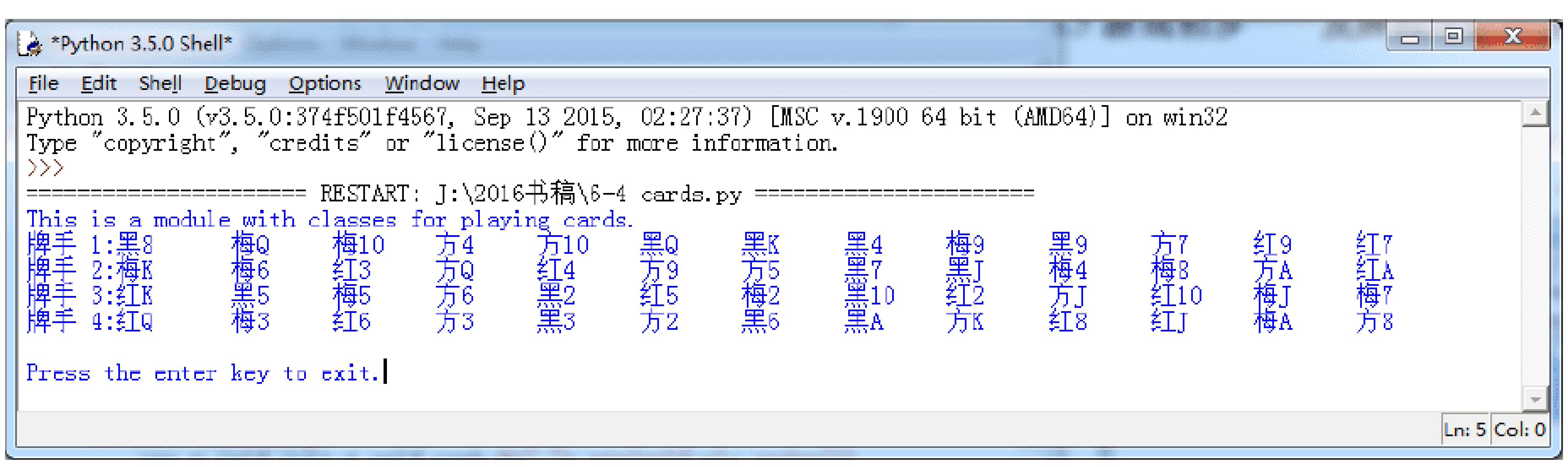


图 1-5 扑克牌发牌程序的运行效果

发牌程序设计出 3 个类——Card 类、Hand 类和 Poke 类。

❶ Card 类

Card 类代表一张牌，其中 FaceNum 字段指的是牌面数字 1～13，Suit 字段指的是花色，值“梅”为梅花、“方”为方钻、“红”为红心、“黑”为黑桃。

（1）Card 构造函数根据参数初始化封装的成员变量，实现牌面大小和花色的初始化，以及是否显示牌面，默认 True 为显示牌的正面。

（2）__str__()方法用来输出牌面大小和花色。

（3）pic_order()方法获取牌的顺序号，牌面按梅花 1～13→方块 14～26→红桃 27～39→黑桃 40～52 的顺序编号（未洗牌之前）。也就是说，梅花 2 的顺序号为 2，方块 A 的

顺序号为 14，方块 K 的顺序号为 26。这个方法是为图形化显示牌面预留的方法。

（4）flip()是翻牌方法，改变牌面是否显示的属性值。

视频讲解

```
#Cards Module
class Card():
    """ A playing card. """
    RANKS=["A", "2", "3", "4", "5", "6", "7",
           "8", "9", "10", "J", "Q", "K"]     #牌面数字 1～13
    SUITS=["梅", "方", "红", "黑"] #"梅"为梅花，"方"为方钻，"红"为红心，"黑"为黑桃

    def __init__(self, rank, suit, face up=True):
        self.rank=rank            #指的是牌面数字 1～13
        self.suit=suit            #指的是花色
        self.is_face_up=face_up  #是否显示牌的正面，True 为正面 False 为背面

    def __str__(self):            #重写 print()方法，打印一张牌的信息
        if self.is face up:
            rep=self.suit+self.rank
        else:
            rep="XX"
        return rep

    def  pic_order(self):         #牌的顺序号
        if self.rank=="A":
            FaceNum=1
        elif self.rank=="J":
            FaceNum=11
        elif self.rank=="Q":
            FaceNum=12
        elif self.rank=="K":
            FaceNum=13
        else:
            FaceNum=int(self.rank)
        if self.suit=="梅":
            Suit=1
        elif self.suit=="方":
            Suit=2
        elif self.suit=="红":
            Suit=3
        else:
            Suit=4
        return(Suit-1)*13+FaceNum

    def flip(self):               #翻牌方法
        self.is_face_up=not self.is_face_up
```

❷ Hand 类

Hand 类代表一手牌（一个玩家手里拿的牌），可以认为是一位牌手手里的牌，其中 cards 列表变量存储牌手手里的牌。玩家可以增加牌、清空手里的牌、把一张牌给其他牌手。

```
class Hand():
    """ A hand of playing cards. """
    def __init__(self):
        self.cards=[]                           #cards 列表变量存储牌手的牌
    def __str__(self):                          #重写 print()方法，打印出牌手的所有牌
        if self.cards:
            rep=""
            for card in self.cards:
                rep+=str(card)+"\t"
        else:
            rep="无牌"
        return rep
    def clear(self):                            #清空手里的牌
        self.cards=[]
    def add(self, card):                        #增加牌
        self.cards.append(card)
    def give(self, card, other_hand):           #把一张牌给其他牌手
        self.cards.remove(card)
        other_hand.add(card)
```

❸ Poke 类

Poke 类代表一副牌，可以看作是有 52 张牌的牌手，所以继承 Hand 类。由于其中 cards 列表变量要存储 52 张牌，而且要发牌、洗牌，所以增加如下方法：

（1）populate(self)生成存储了 52 张牌的一手牌，当然这些牌是按梅花 1～13→方块 14～26→红桃 27～39→黑桃 40～52 的顺序（未洗牌之前）存储在 cards 列表变量中。

（2）shuffle(self)洗牌，使用 Python 的 random 模块的 shuffle()方法打乱牌的存储顺序即可。

（3）deal(self, hands, per_hand=13)是完成发牌动作，发给 4 个玩家，每人默认 13 张牌。当然，如果给 per_hand 传 10，则每人发 10 张牌，只不过牌没发完而已。

```
#Poke 类
class Poke(Hand):
    """ A deck of playing cards. """
    def populate(self):                         #生成一副牌
        for suit in Card.SUITS:
            for rank in Card.RANKS:
                self.add(Card(rank, suit))
    def shuffle(self):                          #洗牌
        import random
        random.shuffle(self.cards)              #打乱牌的顺序
```

```
    def deal(self, hands, per_hand=13):      #发牌，发给玩家，每人默认 13 张牌
        for rounds in range(per_hand):
            for hand in hands:
                if self.cards:
                    top_card=self.cards[0]
                    self.cards.remove(top_card)
                    hand.add(top_card)
                    #self.give(top_card, hand)  #上两句可以用此语句替换
                else:
                    print("不能继续发牌了，牌已经发完！")
```

❹ 主程序

主程序比较简单，因为有 4 个玩家，所以生成 players 列表存储初始化的 4 位牌手。生成一副牌对象实例 poke1，调用 populate()方法生成有 52 张牌的一副牌，调用 shuffle()方法洗牌打乱顺序，调用 deal(players,13)方法发给玩家每人 13 张牌，最后显示 4 位牌手所有的牌。

```
#主程序
if __name__=="__main__":
    print("This is a module with classes for playing cards.")
    #4 个玩家
    players=[Hand(),Hand(),Hand(),Hand()]
    poke1=Poke()
    poke1.populate()                #生成一副牌
    poke1.shuffle()                 #洗牌
    poke1.deal(players,13)          #发给玩家每人 13 张牌
    #显示 4 位牌手的牌
    n=1
    for hand in players:
        print("牌手",n,end=":")
        print(hand)
        n=n+1
    input("\nPress the enter key to exit.")
```

1.4 Python 图形界面设计

Python 提供了多个图形开发界面的库，几个常用的 Python GUI 库如下。

（1）Tkinter：Tkinter 模块（Tk 接口）是 Python 的标准 Tk GUI 工具包的接口。Tkinter 可以在大多数 Unix 平台下使用，同样可以应用在 Windows 和 Macintosh 系统里。Tk8.0 的后续版本可以实现本地窗口风格，并良好地运行在绝大多数平台中。

（2）wxPython：wxPython 是一款开源软件，是 Python 语言的一套优秀的 GUI 图形库，允许 Python 程序员很方便地创建完整的、功能键全的 GUI 用户界面。

（3）Jython：Jython 程序可以和 Java 无缝集成。除了一些标准模块以外，Jython 使用 Java 的模块。Jython 几乎拥有标准的 Python 中不依赖于 C 语言的全部模块。例如，Jython 的用户界面使用 Swing、AWT 或者 SWT。Jython 可以被动态或静态地编译成 Java 字节码。

Tkinter 是 Python 的标准 GUI 库。由于 Tkinter 是内置到 Python 的安装包中，只要安装好 Python，之后就能 import Tkinter 库，而且 IDLE 也是用 Tkinter 编写而成，对于简单的图形界面，Tkinter 还是能应付自如的，使用 Tkinter 可以快速地创建 GUI 应用程序。本书主要使用 Tkinter 设计图形界面。

视频讲解

1.4.1　创建 Windows 窗口

例 1-13　使用 Tkinter 创建一个 Windows 窗口的 GUI 程序。

```
import tkinter                          #导入 Tkinter 模块
win=tkinter.Tk()                        #创建 Windows 窗口对象
win.title('我的第一个 GUI 程序')          #设置窗口标题
win.mainloop()                          #进入消息循环，也就是显示窗口
```

以上代码的执行结果如图 1-6 所示，可见 Tkinter 可以很方便地创建 Windows 窗口。

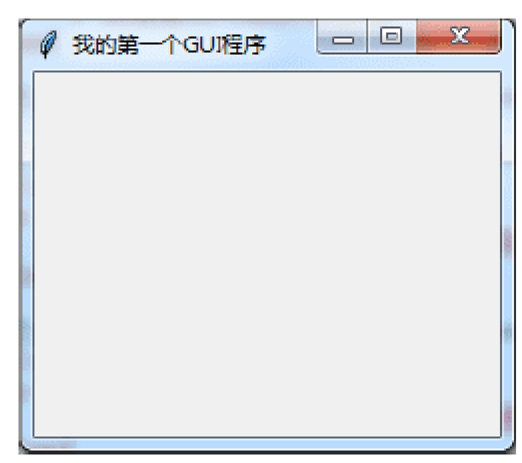


图 1-6　Tkinter 创建一个窗口

在创建 Windows 窗口对象之后，可以使用 geometry()方法设置窗口的大小，格式如下：

```
窗口对象.geometry("size")
```

size 用于指定窗口大小，格式如下：

```
宽度 x 高度    （注：x 是小写字母，不是乘号）
```

例 1-14　显示一个 Windows 窗口，初始大小为 800×600。

```
from tkinter import *
win=Tk();
win.geometry("800x600")
win.mainloop();
```

另外，还可以使用 minsize()方法设置窗口的最小尺寸，使用 maxsize()方法设置窗口的最大尺寸，方法如下：

```
窗口对象. minsize("最小宽度 x 最小高度")
窗口对象. maxsize("最大宽度 x 最大高度")
```

例如：

```
win. minsize("400x600")
win. maxsize("1440x800")
```

1.4.2　几何布局管理器

Tkinter 几何布局管理器（Geometry Manager）用于组织和管理父组件（往往是窗口）

中子组件的布局方式。Tkinter 提供了 3 种不同风格的几何布局管理类，即 pack、grid 和 place。

❶ pack 几何布局管理器

pack 几何布局管理器采用块的方式组织组件，pack 布局根据子组件创建生成的顺序将其放在快速生成的界面中。

调用子组件的方法 pack()，则该子组件在其父组件中采用 pack 布局：

```
pack(option=value,...)
```

pack()方法提供了如表 1-1 所示的若干参数选项。

表 1-1　pack()方法提供的参数选项

选　　项	描　　述	取 值 范 围
side	停靠在父组件的哪一边上	'top'（默认值）、'bottom'、'left'、'right'
anchor	停靠位置，对应于东、南、西、北以及 4 个角	'n'、's'、'e'、'w'、'nw'、'sw'、'se'、'ne'、'center'（默认值）
fill	填充空间	'x'、'y'、'both'、'none'
expand	扩展空间	0 或 1
ipadx、ipady	组件内部在 x/y 方向上填充的空间大小	单位为 c（厘米）、m（毫米）、i（英寸）、p（打印机的点）
padx、pady	组件外部在 x/y 方向上填充的空间大小	单位为 c（厘米）、m（毫米）、i（英寸）、p（打印机的点）

例 1-15　pack 几何布局管理器的 GUI 程序，运行效果如图 1-7 所示。

```
import tkinter
root=tkinter.Tk()
label=tkinter.Label(root,text='hello,python')
label.pack()                          #将 Label 组件添加到窗口中显示
button1=tkinter.Button(root,text='BUTTON1')
                                      #创建文字是“BUTTON1”的 Button 组件
button1.pack(side=tkinter.LEFT)
                                      #将 button1 组件添加到窗口中显示，左停靠
button2=tkinter.Button(root,text='BUTTON2')
                                      #创建文字是“BUTTON2”的 Button 组件
button2.pack(side=tkinter.RIGHT)
                                      #将 button2 组件添加到窗口中显示，右停靠
root.mainloop()
```

❷ grid 几何布局管理器

grid 几何布局管理器采用表格结构组织组件。子组件的位置由行/列确定的单元格决定，子组件可以跨越多行/列。在每一列中，列宽由这一列中最宽的单元格确定。grid 几何布局管理器适合表现表格形式的布局，可以实现复杂的界面，因而被广泛采用。

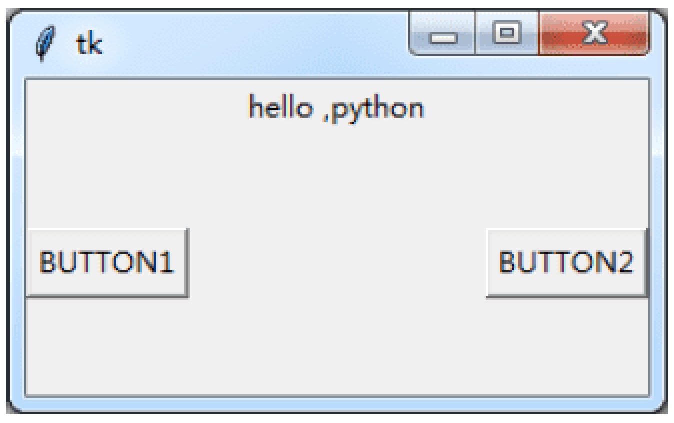


图 1-7　pack 几何布局管理器

调用子组件的 grid()方法，则该子组件在其父组件中采用 grid 几何布局：

```
grid(option=value,...)
```

grid()方法提供了如表 1-2 所示的若干参数选项。

表 1-2　grid()方法提供的参数选项

选　　项	描　　述	取 值 范 围
sticky	组件紧贴所在单元格的某一边角，对应于东、南、西、北以及 4 个角	'n'、's'、'e'、'w'、'nw'、'sw'、'se'、'ne'、'center'（默认值）
row	单元格行号	整数
column	单元格列号	整数
rowspan	行跨度	整数
columnspan	列跨度	整数
ipadx、ipady	组件内部在 x/y 方向上填充的空间大小	单位为 c（厘米）、m（毫米）、i（英寸）、p（打印机的点）
padx、pady	组件外部在 x/y 方向上填充的空间大小	单位为 c（厘米）、m（毫米）、i（英寸）、p（打印机的点）

grid 几何布局管理器有两个最为重要的参数，一个是 row，另一个是 column，用来指定将子组件放置到什么位置，如果不指定 row，会将子组件放置到第 1 个可用的行上，如果不指定 column，则使用第 0 列（首列）。

例 1-16　grid 几何布局管理器的 GUI 程序，运行效果如图 1-8 所示。

```
from tkinter import *
root=Tk()
#200x200 代表了初始化时主窗口的大小，280 和 280 代表了初始化时窗口所在的位置
root.geometry('200x200+280+280')
root.title('计算器示例')
#grid(网格)布局
L1=Button(root, text='1', width=5, bg='yellow')
L2=Button(root, text='2', width=5)
L3=Button(root, text='3', width=5)
L4=Button(root, text='4', width=5)
L5=Button(root, text='5', width=5, bg='green')
L6=Button(root, text='6', width=5)
L7=Button(root, text='7', width=5)
L8=Button(root, text='8', width=5)
L9=Button(root, text='9', width=5, bg='yellow')
L0=Button(root, text='0')
Lp=Button(root, text='.')
L1.grid(row=0, column=0)    #按钮放置在 0 行 0 列
L2.grid(row=0, column=1)    #按钮放置在 0 行 1 列
L3.grid(row=0, column=2)    #按钮放置在 0 行 2 列
L4.grid(row=1, column=0)    #按钮放置在 1 行 0 列
L5.grid(row=1, column=1)    #按钮放置在 1 行 1 列
L6.grid(row=1, column=2)    #按钮放置在 1 行 2 列
L7.grid(row=2, column=0)    #按钮放置在 2 行 0 列
```

```
L8.grid(row=2, column=1)     #按钮放置在 2 行 1 列
L9.grid(row=2, column=2)     #按钮放置在 2 行 2 列
L0.grid(row=3, column=0, columnspan=2,sticky=E+W)     #跨两列，左右贴紧
Lp.grid(row=3, column=2,sticky=E+W)                   #左右贴紧
root.mainloop()
```



图 1-8　grid 几何布局管理器

❸ place 几何布局管理器

place 几何布局管理器允许指定组件的大小与位置。place 几何布局管理器的优点是可以精确地控制组件的位置，不足之处是改变窗口大小时子组件不能随之灵活地改变大小。

调用子组件的方法 place()，则该子组件在其父组件中采用 place 布局：

```
place(option=value,…)
```

place()方法提供了如表 1-3 所示的若干参数选项，用户可以直接给参数选项赋值加以修改。

表 1-3　place()方法提供的参数选项

选　　项	描　　述	取 值 范 围
x,y	将组件放到指定位置的绝对坐标	从 0 开始的整数
relx, rely	将组件放到指定位置的相对坐标	0～1.0
height, width	高度和宽度，单位为像素	
anchor	对齐方式，对应于东、南、西、北以及 4 个角	'n'、's'、'e'、'w'、'nw'、'sw'、'se'、'ne'、'center'（默认值）

注意：Python 的坐标系是左上角为原点位置(0,0)，向右是 x 坐标正方向，向下是 y 坐标正方向，这和数学中的几何坐标系不同，大家一定要注意这一点。

例 1-17　place 几何布局管理器的 GUI 示例程序，运行效果如图 1-9 所示。

```
from tkinter import *
root=Tk()
root.title("登录")
root['width']=200;root['height']=80
Label(root,text='用户名',width=6).place(x=1,y=1)          #绝对坐标(1,1)
Entry(root,width=20).place(x=45,y=1)                      #绝对坐标(45,20)
Label(root,text='密码',width=6).place(x=1,y=20)           #绝对坐标(1,20)
Entry(root,width=20,show='*').place(x=45,y=20)            #绝对坐标(45,20)
Button(root,text='登录',width=8).place(x=40,y=40)         #绝对坐标(40,40)
Button(root,text='取消',width=8).place(x=110,y=40)        #绝对坐标(110,40)
root.mainloop()
```

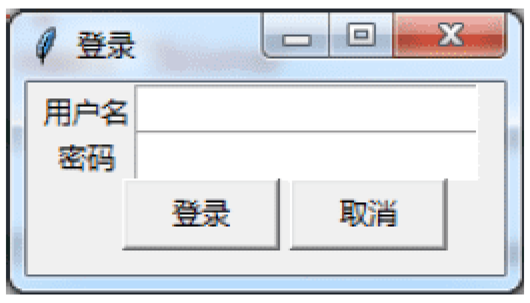


图 1-9　place 几何布局管理器

1.4.3　Tkinter 组件

Tkinter 提供了很多组件，例如按钮、标签和文本框等，在一个 GUI 应用程序中使用，这些组件通常被称为控件或者部件。Tkinter 组件如表 1-4 所示。

表 1-4　Tkinter 组件

控　件	描　述
Button	按钮控件，在程序中显示按钮
Canvas	画布控件，显示图形元素，例如线条或文本
Checkbutton	多选框控件，用于在程序中提供多项选择框
Entry	输入控件，用于显示简单的文本内容
Frame	框架控件，在屏幕上显示一个矩形区域，多用来作为容器
Label	标签控件，可以显示文本和位图
Listbox	列表框控件，用来显示一个字符串列表给用户
Menubutton	菜单按钮控件，用于显示菜单项
Menu	菜单控件，显示菜单栏、下拉菜单和弹出菜单
Message	消息控件，用来显示多行文本，与 Label 比较类似
Radiobutton	单选按钮控件，显示一个单选的按钮状态
Scale	范围控件，显示一个数值刻度，为输出限定范围的数字区间
Scrollbar	滚动条控件，在内容超过可视化区域时使用，例如列表框
Text	文本控件，用于显示多行文本
Toplevel	容器控件，用来提供一个单独的对话框，和 Frame 比较类似
Spinbox	输入控件，与 Entry 类似，但是可以指定输入范围值
PanedWindow	PanedWindow 是一个窗口布局管理的插件，可以包含一个或者多个子控件
LabelFrame	LabelFrame 是一个简单的容器控件，常用于复杂的窗口布局
tkMessageBox	用于显示应用程序的消息框

通过组件类的构造函数可以创建其对象实例。例如：

```
from tkinter import *
root=Tk()
button1=Button(root, text="确定")            #按钮组件的构造函数
```

组件的标准属性也就是所有组件（控件）的共同属性，例如大小、字体和颜色等。Tkinter 组件常用的标准属性如表 1-5 所示。

表 1-5　Tkinter 组件常用的标准属性

属　性	描　述
dimension	控件大小
color	控件颜色
font	控件字体
anchor	锚点（内容停靠位置），对应于东、南、西、北以及 4 个角
relief	控件样式
bitmap	位图，内置位图包括 error、gray75、gray50、gray25、gray12、info、questhead、hourglass、question 和 warning，自定义位图为.xbm 格式的文件

续表

属　　性	描　　述
cursor	光标
text	显示文本内容
state	设置组件状态为正常（normal）、激活（active）或禁用（disabled）

用户可以通过下列方式之一设置组件的属性。

```
button1=Button(root, text="确定")     #按钮组件的构造函数
button1. config(text="确定")          #组件对象的 config()方法的命名参数
button1 ["text"]="确定"               #组件对象的属性的赋值
```

❶ 标签组件 Label

Label 组件用于在窗口中显示文本或位图。anchor 属性指定文本（Text）或图像（Bitmap/Image）在 Label 中的显示位置（如图 1-10 所示，其他组件同此），对应于东、南、西、北以及 4 个角，可用值如下。

- e：垂直居中，水平居右。
- w：垂直居中，水平居左。
- n：垂直居上，水平居中。
- s：垂直居下，水平居中。
- ne：垂直居上，水平居右。
- se：垂直居下，水平居右。
- sw：垂直居下，水平居左。
- nw：垂直居上，水平居左。
- center（默认值）：垂直居中，水平居中。

例 1-18　Label 组件示例，运行效果如图 1-11 所示。

```
from tkinter import *
win=Tk();                                           #创建窗口对象
win.title("我的窗口")                                #设置窗口标题
lab1=Label(win,text='你好', anchor='nw')            #创建文字是“你好”的 Label 组件
lab1.pack()                                         #显示 Label 组件
#显示内置的位图
lab2=Label(win, bitmap='question')                  #创建显示疑问图标 Label 组件
lab2.pack()                                         #显示 Label 组件

#显示自选的图片
bm=PhotoImage(file=r'J:\2018 书稿\aa.png')
lab3=Label(win,image=bm)
lab3.bm=bm
lab3.pack()                                         #显示 Label 组件
win.mainloop()
```

❷ 按钮组件 Button

Button 组件（控件）是一个标准的 Tkinter 部件，用于实现各种按钮。按钮可以包含文本或图像，可以通过 command 属性将调用函数或方法关联到按钮上。当 Tkinter 的按钮被

按下时会自动调用该函数或方法。

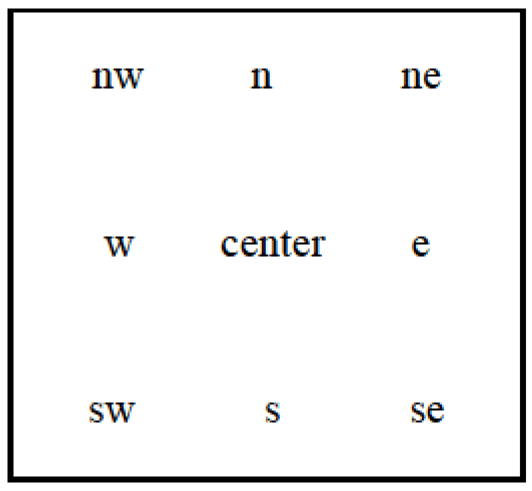


图 1-10　anchor 地理方位



图 1-11　Label 组件示例

❸ 单行文本框组件 Entry 和多行文本框组件 Text

Entry 组件主要用于输入单行内容和显示文本，可以方便地向程序传递用户参数。这里通过一个转换摄氏度和华氏度的小程序来演示该组件的使用。

1）创建和显示 Entry 对象

创建 Entry 对象的基本方法如下：

```
Entry 对象=Entry(Windows 窗口对象)
```

显示 Entry 对象的方法如下：

```
Entry 对象.pack()
```

2）获取 Entry 组件的内容

get()方法用于获取单行文本框内输入的内容。

设置或者获取 Entry 组件内容也可以使用 StringVar()对象来完成，把 Entry 的 textvariable 属性设置为 StringVar()变量，再通过 StringVar()变量的 get()和 set()函数读取和输出相应文本内容。例如：

```
s=StringVar()                                #一个 StringVar()对象
s.set("大家好，这是测试")
entryCd=Entry(root, textvariable=s)          #Entry 组件显示“大家好，这是测试”
print(s.get())                               #打印出“大家好，这是测试”
```

3）Entry 的常用属性

- show：如果设置为字符*，则输入文本框内的显示为*，用于密码输入。
- insertbackground：插入光标的颜色，默认为'black'。
- selectbackground 和 selectforeground：选中文本的背景色与前景色。
- width：组件的宽度（所占字符个数）。
- fg：字体的前景颜色。
- bg：背景颜色。
- state：设置组件状态，默认为 normal，还可设置为 disabled（禁用组件）或 readonly（只读）。

Python 还提供了多行文本框组件 Text，用于输入多行内容和显示文本。其使用方法类

似 Entry，请读者参考 Tkinter 手册。

❹ 列表框组件 Listbox

列表框组件 Listbox 用于显示多个项目，并且允许用户选择一个或多个项目。

1）创建和显示 Listbox 对象

创建 Listbox 对象的基本方法如下：

```
Listbox 对象=Listbox(Tkinter Windows 窗口对象)
```

显示 Listbox 对象的方法如下：

```
Listbox 对象.pack()
```

2）插入文本项

用户可以使用 insert()方法向列表框组件中插入文本项，方法如下：

```
Listbox 对象.insert(index,item)
```

其中，index 是插入文本项的位置，如果在尾部插入文本项，则可以使用 END；如果在当前选中处插入文本项，则可以使用 ACTIVE。item 是要插入的文本项。

3）返回选中项目的索引

```
Listbox 对象.curselection()
```

返回当前选中项目的索引，结果为元组。

注意：索引号从 0 开始，0 表示第 1 项。

4）删除文本项

```
Listbox 对象.delete(first,last)
```

删除指定范围 first～last 的项目，当不指定 last 时删除 1 个项目。

5）获取项目内容

```
Listbox 对象.get(first,last)
```

返回指定范围 first～last 的项目，当不指定 last 时仅返回 1 个项目。

6）获取项目个数

```
Listbox 对象.size()
```

7）获取 Listbox 内容

需要使用 listvariable 属性为 Listbox 对象指定一个对应的变量，例如：

```
m=StringVar()
listb=Listbox(root, listvariable=m)
listb.pack()
root.mainloop()
```

指定后就可以使用 m.get()方法获取 Listbox 对象中的内容了。

注意：如果允许用户选择多个项目，需要将 Listbox 对象的 selectmode 属性设置为

MULTIPLE（表示多选），而设置为 SINGLE 表示单选。

例 1-19　创建从一个列表框选择内容添加到另一个列表框的 GUI 程序。

```
from tkinter import *                    #导入 Tkinter 库
root=Tk()                                #创建窗口对象
def callbutton1():
    for i in listb.curselection():       #遍历选中项
        listb2.insert(0,listb.get(i))    #添加到右侧列表框

def callbutton2():
    for i in listb2.curselection():      #遍历选中项
        listb2.delete(i)                 #从右侧列表框中删除
#创建两个列表
li=['C','python','php','html','SQL','java']
listb=Listbox(root)                      #创建两个列表框组件
listb2=Listbox(root)
for item in li:                          #左侧列表框组件插入数据
    listb.insert(0,item)
listb.grid(row=0,column=0,rowspan=2)     #将列表框组件放置到窗口对象中
b1=Button(root,text='添加>>', command=callbutton1, width=20)
                                         #创建 Button 组件
b2=Button(root,text='删除<<', command=callbutton2, width=20)
                                         #创建 Button 组件
b1.grid(row=0,column=1,rowspan=2)        #显示 Button 组件
b2.grid(row=1,column=1,rowspan=2)        #显示 Button 组件
listb2.grid(row=0,column=2,rowspan=2)
root.mainloop()                          #进入消息循环
```

以上代码的执行结果如图 1-12 所示。

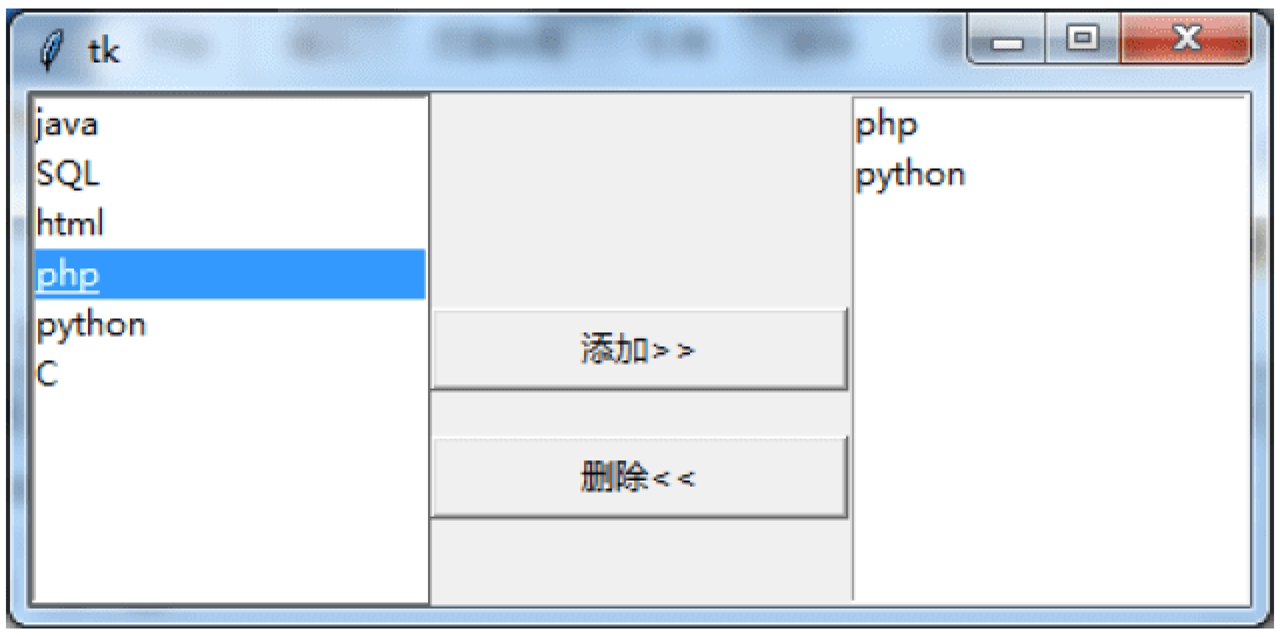


图 1-12　含有两个列表框组件的 GUI 程序

❺ 单选按钮组件 Radiobutton 和复选框组件 Checkbutton

单选按钮组件 Radiobutton 和复选框组件 Checkbutton 分别用于实现选项的单选和复选功能。Radiobutton 用于从同一组单选按钮中选择一个单选按钮（不能同时选择多个）。Radiobutton 可以显示文本，也可以显示图像。Checkbutton 用于选择一项或多项，同样

Checkbutton 可以显示文本，也可以显示图像。

1）创建和显示 Radiobutton 对象

创建 Radiobutton 对象的基本方法如下：

```
Radiobutton 对象=Radiobutton(Windows 窗口对象,text=Radiobutton 组件显示的文本)
```

显示 Radiobutton 对象的方法如下：

```
Radiobutton 对象.pack()
```

用户可以使用 variable 属性为 Radiobutton 组件指定一个对应的变量。如果将多个 Radiobutton 组件绑定到同一个变量，则这些 Radiobutton 组件属于一个分组。分组后需要使用 value 设置每个 Radiobutton 组件的值，以标识该项目是否被选中。

2）Radiobutton 组件的常用属性

- variable：单选按钮索引变量，通过变量的值确定哪个单选按钮被选中。一组单选按钮使用同一个索引变量。
- value：单选按钮选中时变量的值。
- command：单选按钮选中时执行的命令（函数）。

3）Radiobutton 组件的方法

- deselect()：取消选择。
- select()：选择。
- invoke()：调用单选按钮 command 指定的回调函数。

4）创建和显示 Checkbutton 对象

创建 Checkbutton 对象的基本方法如下：

```
Checkbutton 对象=Checkbutton(Tkinter Windows 窗口对象,text=Checkbutton 组件显示的文本, command=单击 Checkbutton 按钮所调用的回调函数)
```

显示 Checkbutton 对象的方法如下：

```
Checkbutton 对象.pack()
```

5）Checkbutton 组件的常用属性

- variable：复选框索引变量，通过变量的值确定哪些复选框被选中。每个复选框使用不同的变量，使复选框之间相互独立。
- onvalue：复选框选中（有效）时变量的值。
- offvalue：复选框未选中（无效）时变量的值。
- command：复选框选中时执行的命令（函数）。

6）获取 Checkbutton 组件的状态

为了获取 Checkbutton 组件是否被选中，需要使用 variable 属性为 Checkbutton 组件指定一个对应变量，例如：

```
c=tkinter.IntVar()
c.set(2)
check=tkinter.Checkbutton(root,text='喜欢',variable=c,onvalue=1,
```

```
offvalue=2)                                     #1 为选中，2 为没选中
check.pack()
```

指定变量 c 后，可以使用 c.get()获取复选框的状态值，也可以使用 c.set()设置复选框的状态。例如设置 check 对象为没选中状态，代码如下：

```
c.set(2)                    #1 为选中，2 为没选中，设置为 2 就表示没选中状态
```

获取单选按钮（Radiobutton）状态的方法同上。

例1-20　创建使用单选按钮（Radiobutton）组件选择国家的程序。

```
import tkinter
root=tkinter.Tk()
r=tkinter.StringVar()                       #创建 StringVar 对象
r.set('1')                                  #设置初始值为“1”，初始选中“中国”
radio=tkinter.Radiobutton(root,variable=r,value='1',text='中国')
radio.pack()
radio=tkinter.Radiobutton(root,variable=r,value='2',text='美国')
radio.pack()
radio=tkinter.Radiobutton(root,variable=r,value='3',text='日本')
radio.pack()
radio=tkinter.Radiobutton(root,variable=r,value='4',text='加拿大')
radio.pack()
radio=tkinter.Radiobutton(root,variable=r,value='5',text='韩国')
radio.pack()
root.mainloop()
print(r.get())                              #获取被选中单选按钮变量值
```

以上代码的执行结果如图 1-13 所示。选中日本后打印出 3。

❻ 菜单组件 Menu

图形用户界面应用程序通常提供菜单，菜单包含各种按照主题分组的基本命令。通常，图形用户界面应用程序包括两种类型的菜单。

（1）主菜单：提供窗体的菜单系统。通过单击可下拉出子菜单，选择命令可执行相关的操作。常用的主菜单一般包括文件、编辑、视图、帮助等。

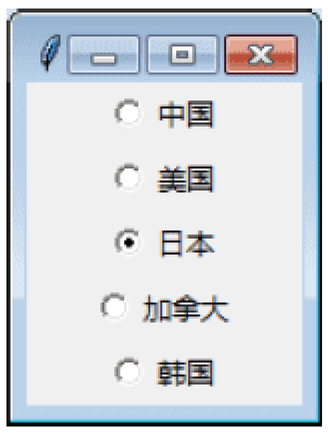


图 1-13　单选按钮示例程序

（2）上下文菜单（也称为快捷菜单）：通过右击某对象而弹出的菜单，一般为与该对象相关的常用菜单命令，例如剪切、复制、粘贴等。

创建 Menu 对象的基本方法如下：

```
Menu 对象=Menu(Windows 窗口对象)
```

将 Menu 对象显示在窗口中的方法如下：

```
Windows 窗口对象['menu']=Menu 对象
Windows 窗口对象.mainloop()
```

例1-21　使用 Menu 组件的简单例子，运行效果如图 1-14 所示。

```
from tkinter import *
root=Tk()
def hello():                              #菜单项事件函数，每个菜单项可以单独写
   print("你单击主菜单")
m=Menu(root)
for item in ['文件','编辑','视图']:       #添加菜单项
   m.add command(label=item, command=hello)
root['menu']=m                            #附加主菜单到窗口
root.mainloop()
```

❼ 消息窗口组件 Messagebox

消息窗口组件 Messagebox 用于弹出提示框向用户进行告警，或让用户选择下一步如何操作。消息框包括很多类型，常用的有 info、warning、error、yesno、okcancel 等，包含不同的图标、按钮以及弹出提示音。

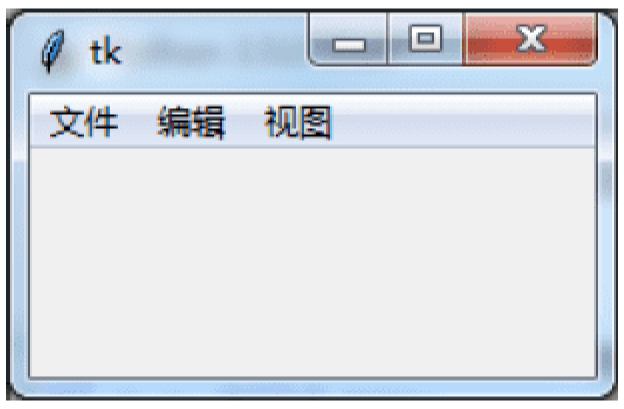


图 1-14　使用 Menu 组件的菜单运行效果

例1-22　演示各消息框的程序，运行效果如图 1-15 所示。

```
import tkinter as tk
from tkinter import messagebox as msgbox
def btn1 clicked():
   msgbox.showinfo("Info", "Showinfo test.")
def btn2 clicked():
   msgbox.showwarning("Warning", "Showwarning test.")
def btn3 clicked():
   msgbox.showerror("Error", "Showerror test.")
def btn4 clicked():
   msgbox.askquestion("Question", "Askquestion test.")
def btn5 clicked():
   msgbox.askokcancel("OkCancel", "Askokcancel test.")
def btn6 clicked():
   msgbox.askyesno("YesNo", "Askyesno test.")
def btn7 clicked():
   msgbox.askretrycancel("Retry", "Askretrycancel test.")
root=tk.Tk()
root.title("MsgBox Test")
btn1=tk.Button(root, text="showinfo", command=btn1 clicked)
btn1.pack(fill=tk.X)
btn2=tk.Button(root, text="showwarning", command=btn2 clicked)
btn2.pack(fill=tk.X)
btn3=tk.Button(root, text="showerror", command=btn3 clicked)
btn3.pack(fill=tk.X)
```

```
btn4=tk.Button(root, text="askquestion", command=btn4_clicked)
btn4.pack(fill=tk.X)
btn5=tk.Button(root, text="askokcancel", command=btn5_clicked)
btn5.pack(fill=tk.X)
btn6=tk.Button(root, text="askyesno", command=btn6_clicked)
btn6.pack(fill=tk.X)
btn7=tk.Button(root, text="askretrycancel", command=btn7_clicked)
btn7.pack(fill=tk.X)
root.mainloop()
```

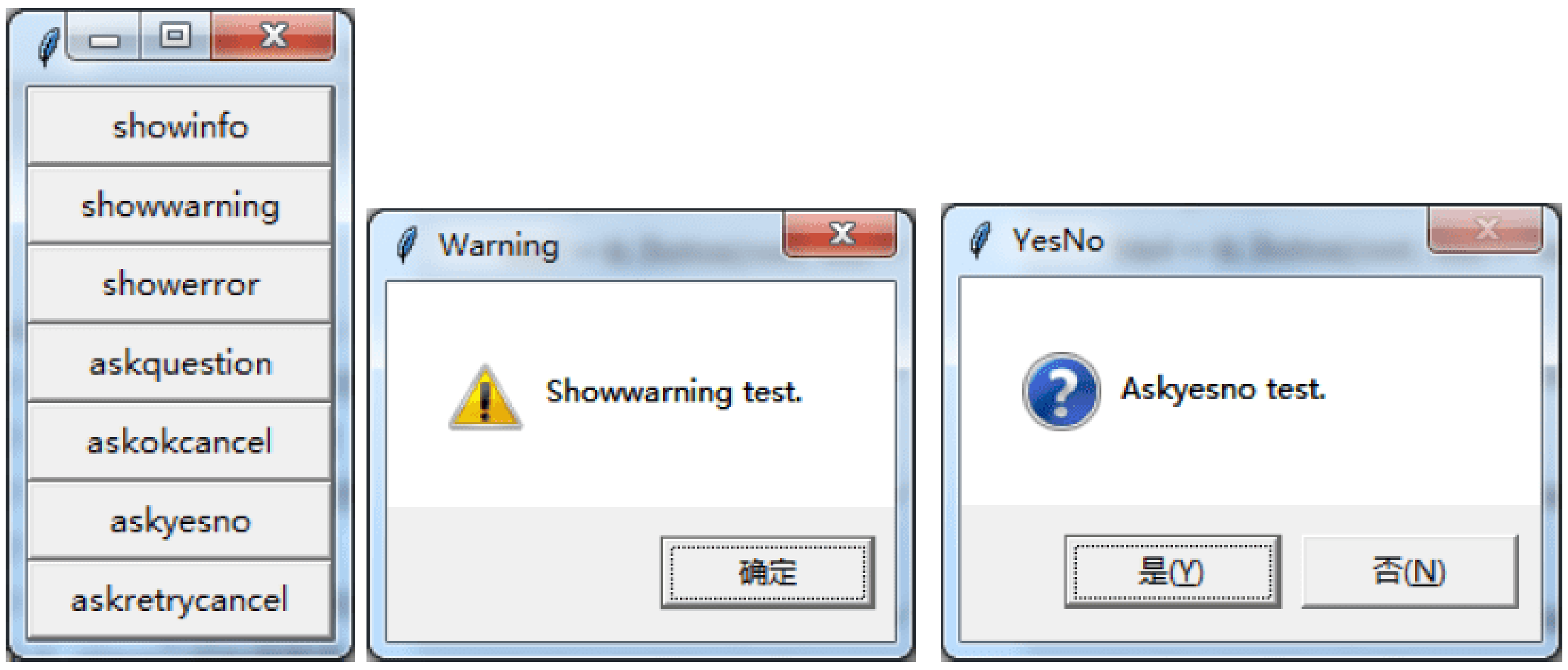


图 1-15　消息窗口的运行效果

❽ 框架组件 Frame

Frame 组件是框架组件，在分组组织其他组件的过程中是非常重要的，负责安排其他组件的位置。Frame 组件在屏幕上显示为一个矩形区域，作为显示其他组件的容器。

1）创建和显示 Frame 对象

创建 Frame 对象的基本方法如下：

```
Frame 对象=Frame(窗口对象, height=高度,width=宽度,bg=背景色, ...)
```

例如，创建第 1 个 Frame 组件，其高为 100，宽为 400，背景色为绿色。

```
f1=Frame(root, height=100,width=400,bg='green')
```

显示 Frame 对象的方法如下：

```
Frame 对象.pack()
```

2）向 Frame 组件中添加组件

在创建组件时指定其容器为 Frame 组件即可，例如：

```
Label(Frame 对象,text='Hello').pack()    #向 Frame 组件中添加一个 Label 组件
```

3）LabelFrame 组件

LabelFrame 组件是有标题的 Frame 组件，可以使用 text 属性设置 LabelFrame 组件的标题，方法如下：

```
LabelFrame(窗口对象, height=高度,width=宽度,text=标题).pack()
```

例 1-23　使用两个 Frame 组件和一个 LabelFrame 组件的例子。

```
from tkinter import *
root=Tk()                                    #创建窗口对象
root.title("使用 Frame 组件的例子")          #设置窗口标题
f1=Frame(root)                               #创建第 1 个 Frame 组件
f1.pack()
f2=Frame(root)                               #创建第 2 个 Frame 组件
f2.pack()
f3=LabelFrame(root,text='第 3 个 Frame')
                                             #第 3 个 LabelFrame 组件，放置在窗口底部
f3.pack(side=BOTTOM)
redbutton=Button(f1, text="Red", fg="red")
redbutton.pack(side=LEFT)
brownbutton=Button(f1, text="Brown", fg="brown")
brownbutton.pack(side=LEFT)
bluebutton=Button(f1, text="Blue", fg="blue")
bluebutton.pack(side=LEFT)
blackbutton=Button(f2, text="Black", fg="black")
blackbutton.pack()
greenbutton=Button(f3, text="Green", fg="Green")
greenbutton.pack()
root.mainloop()
```

以上代码通过 Frame 框架把 5 个按钮分成 3 个区域，第 1 个区域 3 个按钮，第 2 个区域 1 个按钮，第 3 个区域 1 个按钮，运行效果如图 1-16 所示。

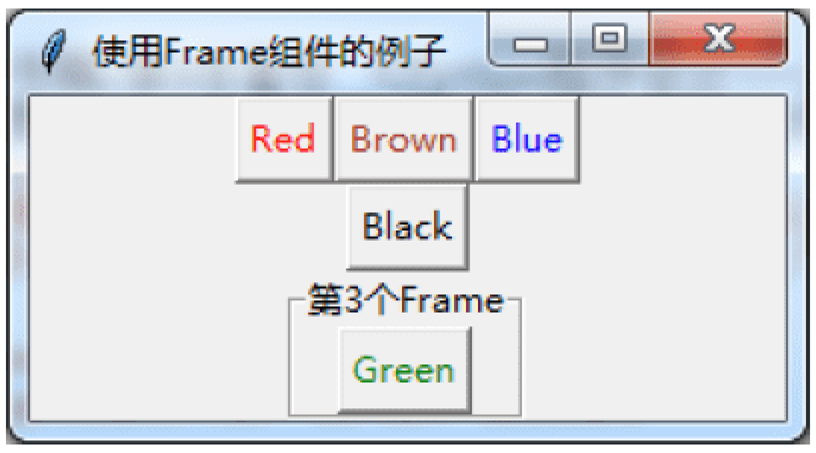


图 1-16　Frame 框架的运行效果

4）刷新 Frame

用 Python 做 GUI 图形界面，可以使用 after()方法每隔几秒刷新 GUI 图形界面。例如下面的代码实现计数器功能，并且文字背景色不断改变。

```
from tkinter import *
colors=('red', 'orange', 'yellow', 'green', 'blue', 'purple')
root=Tk()
f=Frame(root, height=200, width=200)
f.color=0
f['bg']=colors[f.color]  #设置框架背景色
lab1=Label(f,text='0')
```

```
lab1.pack()
def foo():
    f.color=(f.color+1)%(len(colors))
    lab1['bg']=colors[f.color]
    lab1['text']=str(int(lab1['text'])+1)
    f.after(500, foo)   #隔 500 ms 执行 foo()函数刷新屏幕
f.pack()
f.after(500, foo)
root.mainloop()
```

例如开发移动电子广告效果就可以使用 after()方法实现不断移动 lab1。

```
from tkinter import *
root=Tk()
f=Frame(root, height=200, width=200)
lab1=Label(f,text='欢迎参观中原工学院')
x=0
def foo():
    global x
    x=x+10
    if x>200:
        x=0
    lab1.place(x=x,y=0)
    f.after(500, foo)   #隔 500 ms 执行 foo()函数刷新屏幕

f.pack()
f.after(500, foo)
root.mainloop()
```

运行程序可见“欢迎参观中原工学院”不停地从左向右移动，出了窗口右侧以后重新从左侧出现。利用此技巧可以开发类似的贪吃蛇游戏，借助 after()方法实现不断改变蛇的位置，从而达到蛇移动的效果。

1.4.4　Tkinter 字体

通过组件的 font 属性可以设置其显示文本的字体，注意在设置组件字体前首先要能表示一个字体。

❶ 通过元组表示字体

通过 3 个元素的元组可以表示字体：

```
(font family,size,modifiers)
```

作为一个元组的第 1 个元素的 font family 是字体名；size 为字体大小，单位为 point；modifiers 为包含粗体、斜体、下画线的样式修饰符。

例如：

```
("Times New Roman ", "16")                #16 点阵的 Times 字体
("Times New Roman ", "24", "bold italic")#24 点阵的 Times 字体，且为粗体、斜体
```

例1-24 通过元组表示字体设置标签的字体，运行效果如图 1-17 所示。

```
from tkinter import *
root=Tk()
#创建 Label
for ft in('Arial',('Courier New',19,'italic'),('Comic Sans MS',),'Fixdsys',
('MS Sans Serif',),('MS Serif',),'Symbol','System',('Times New Roman',),'Verdana'):
    Label(root,text='hello sticky',font=ft).grid()
root.mainloop()
```

这个程序在 Windows 上测试字体显示，注意字体中包含有空格的字体名称必须指定为元组类型。



图 1-17 缩放图形对象运行效果

❷ 通过 Font 对象表示字体

使用 tkFont.Font 来创建字体，格式如下：

```
ft=tkFont.Font(family='字体名',size, weight,
slant, underline, overstrike)
```

其中，size 为字体大小；weight='bold'或'normal'，'bold'为粗体；slant='italic'或'normal'，'italic'为斜体；underline=1 或 0，1 为下画线；overstrike=1 或 0，1 为删除线。

```
ft=Font(family='Helvetica',size=36,weight='bold')
```

例1-25 通过 Font 对象设置标签字体，运行效果如图 1-18 所示。

```
#通过 Font 对象来创建字体
from tkinter import *
import tkinter.font                                    #引入字体模块
root=Tk()
#指定字体名称、大小、样式
ft=tkinter.font.Font(family='Fixdsys',size=20,weight='bold')
Label(root,text='hello sticky',font=ft).grid() #创建一个 Label
root.mainloop()
```

通过 tkFont. families()函数可以返回所有可用的字体。

```
from tkinter import *
import tkinter.font         #引入字体模块
root=Tk()
print(tkinter.font.families())
```

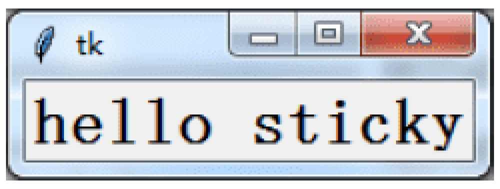


图 1-18 通过 Font 对象设置标签字体

输出结果：

```
('Forte', 'Felix Titling', 'Eras Medium ITC', 'Eras Light ITC', 'Eras Demi
ITC', 'Eras Bold ITC', 'Engravers MT', 'Elephant', 'Edwardian Script ITC',
'Curlz MT', 'Copperplate Gothic Light', 'Copperplate Gothic Bold', 'Century
Schoolbook', 'Castellar', 'Calisto MT', 'Bookman Old Style', 'Bodoni MT
Condensed', 'Bodoni MT Black', 'Bodoni MT', 'Blackadder ITC', 'Arial Rounded
MT Bold', 'Agency FB', 'Bookshelf Symbol 7', 'MS Reference Sans Serif', 'MS
```

```
Reference Specialty', 'Berlin Sans FB Demi', 'Tw Cen MT Condensed Extra Bold',
'Calibri Light', 'Bitstream Vera Sans Mono', '方正兰亭超细黑简体', '@方正兰亭
超细黑简体', 'Buxton Sketch', 'Segoe Marker', 'SketchFlow Print')
```

1.4.5　Python 事件处理

视频讲解

所谓事件（Event），就是程序上发生的事。例如用户敲击键盘上的某一个键或是单击、移动鼠标。对于这些事件，程序需要做出反应。Tkinter 提供的组件通常都有自己可以识别的事件。例如当按钮被单击时执行特定操作，或者当一个输入栏成为焦点，而用户又敲击了键盘上的某些键时，用户所输入的内容就会显示在输入栏内。

程序可以使用事件处理函数来指定当触发某个事件时所做的反应（操作）。

❶ 事件类型

事件类型的通用格式如下：

```
<[modifier-]…type[-detail]>
```

事件类型必须放置于尖括号<>内。type 描述了类型，例如键盘按键、鼠标单击。modifier 用于组合键定义，例如 Control、Alt。detail 用于明确定义是哪一个键或按钮的事件，例如 1 表示鼠标左键、2 表示鼠标中键、3 表示鼠标右键。

举例如下：

```
<Button-1>                         #按下鼠标左键
<KeyPress-A>                       #按下键盘上的 A 键
<Control-Shift-KeyPress-A>         #同时按下了 Control、Shift、A 3 个键
```

在 Python 中，键盘事件见表 1-6，鼠标事件见表 1-7，窗体事件见表 1-8。

表 1-6　键盘事件

名　称	描　述
KeyPress	按下键盘上的某键时触发，可以在 detail 部分指定是哪个键
KeyRelease	释放键盘上的某键时触发，可以在 detail 部分指定是哪个键

表 1-7　鼠标事件

名　称	描　述
ButtonPress 或 Button	按下鼠标某键，可以在 detail 部分指定是哪个键
ButtonRelease	释放鼠标某键，可以在 detail 部分指定是哪个键
Motion	点中组件的同时拖曳组件移动时触发
Enter	当鼠标指针移进某组件时触发
Leave	当鼠标指针移出某组件时触发
MouseWheel	当鼠标滚轮滚动时触发

表 1-8　窗体事件

名　称	描　述
Visibility	当组件变为可视状态时触发

续表

名　　称	描　　述
Unmap	当组件由显示状态变为隐藏状态时触发
Map	当组件由隐藏状态变为显示状态时触发
Expose	当组件从原本被其他组件遮盖的状态中暴露出来时触发
FocusIn	当组件获得焦点时触发
FocusOut	当组件失去焦点时触发
Configure	当改变组件大小时触发，例如拖曳窗体边缘
Property	当窗体的属性被删除或改变时触发，属于 Tk 的核心事件
Destroy	当组件被销毁时触发
Activate	与组件选项中的 state 项有关，表示组件由不可用转为可用，例如按钮由 disabled（灰色）转为 enabled
Deactivate	与组件选项中的 state 项有关，表示组件由可用转为不可用，例如按钮由 enabled 转为 disabled（灰色）

modifier 组合键定义中常用的修饰符见表 1-9。

表 1-9　组合键定义中常用的修饰符

修　饰　符	描　　述
Alt	Alt 键按下
Any	任何按键按下，例如<Any-KeyPress>
Control	Control 键按下
Double	两个事件在短时间内发生，例如双击鼠标左键<Double-Button-1>
Lock	Caps Lock 键按下
Shift	Shift 键按下
Triple	类似于 Double，3 个事件短时间内发生

可以用短格式表示事件，例如<1>等同于<Button-1>、<x>等同于<KeyPress-x>。

对于大多数的单字符按键，用户还可以忽略“<>”符号，但是空格键和尖括号键不能这样做（正确的表示分别为<space>、<less>）。

❷ 事件绑定

程序建立一个处理某一事件的事件处理函数称为绑定。

1）创建组件对象时指定

在创建组件对象实例时，可以通过其命名参数 command 指定事件处理函数。例如：

```
def callback():                                    #事件处理函数
    showinfo("Python command","人生苦短，我用 Python")
Bu1=Button(root, text="设置 command 事件调用命令",command=callback)
Bu1.pack()
```

2）实例绑定

调用组件对象实例方法 bind()可以为指定组件实例绑定事件，这是最常用的事件绑定方式。

```
组件对象实例名.bind("<事件类型>", 事件处理函数)
```

例如，假如声明了一个名为 canvas 的 Canvas 组件对象，想在 canvas 上按下鼠标左键

时画一条线，可以这样实现：

```
canvas.bind("<Button-1>", drawline)
```

其中，bind 函数的第 1 个参数是事件描述符，指定无论什么时候在 canvas 上，当按下鼠标左键时就调用事件处理函数 drawline 进行画线的任务。特别需要说明的是，drawline 后面的圆括号是省略的，Tkinter 会将此函数填入相关参数后调用运行，在这里只是声明而已。

3）标识绑定

在 Canvas 画布中绘制各种图形，将图形与事件绑定可以使用标识绑定函数 tag_bind()。预先为图形定义标识 tag 后，通过标识 tag 来绑定事件。例如：

```
cv.tag_bind('r1','<Button-1>',printRect)
```

例 1-26　标识绑定的例子。

```
from tkinter import *
root=Tk()
def printRect(event):
    print('rectangle 左键事件')
def printRect2(event):
    print('rectangle 右键事件')
def printLine(event):
    print('Line 事件')

cv=Canvas(root,bg='white')          #创建一个 Canvas，设置其背景色为白色
rt1=cv.create rectangle(
    10,10,110,110,
    width=8, tags='r1')
cv.tag_bind('r1','<Button-1>',printRect)     #绑定 item 与鼠标左键事件
cv.tag bind('r1','<Button-3>',printRect2)    #绑定 item 与鼠标右键事件
#创建一个 line，并将其 tags 设置为'r2'
cv.create line(180,70,280,70,width=10,tags='r2')
cv.tag_bind('r2','<Button-1>',printLine)     #绑定 item 与鼠标左键事件
cv.pack()
root.mainloop()
```

在这个示例中，单击到矩形的边框时才会触发事件，矩形既响应鼠标左键又响应右键。当用鼠标左键单击矩形边框时出现“rectangle 左键事件”信息，用鼠标右键单击矩形边框时出现“rectangle 右键事件”信息，用鼠标左键单击直线时出现“Line 事件”信息。

❸ 事件处理函数

1）定义事件处理函数

事件处理函数往往带有一个 event 参数，在触发事件调用事件处理函数时将传递 Event 对象实例。

```
def callback(event):                              #事件处理函数
    showinfo("Python command","人生苦短，我用 Python")
```

2）Event 对象的参数

Event 对象可以获取各种相关参数，主要参数如表 1-10 所示。

表 1-10　Event 对象的主要参数

参　　数	说　　明
.x,.y	鼠标相对于组件对象左上角的坐标
.x_root,.y_root	鼠标相对于屏幕左上角的坐标
.keysym	字符串命名按键，例如 Escape、F1～F12、Scroll_Lock、Pause、Insert、Delete、Home、Prior（这个是 page up）、Next（这个是 page down）、End、Up、Right、Left、Down、Shift_L、Shift_R、Control_L、Control_R、Alt_L、Alt_R、Win_L
.keysym_num	数字代码命名按键
.keycode	键码，但是它不能反映事件前缀 Alt、Control、Shift、Lock，并且它不区分大小写按键，即输入 a 和 A 是相同的键码
.time	时间
.type	事件类型
.widget	触发事件的对应组件
.char	字符

Event 对象按键的详细信息如表 1-11 所示。

表 1-11　Event 对象按键的详细信息

.keysym	.keycode	.keysym_num	说　　明
Alt_L	64	65513	左手边的 Alt 键
Alt_R	113	65514	右手边的 Alt 键
BackSpace	22	65288	BackSpace 键
Cancel	110	65387	Pause Break 键
F1～F11	67～77	65470～65480	功能键 F1～F11
Print	111	65377	打印屏幕键

例 1-27　触发 keyPress 键盘事件的例子，运行效果如图 1-19 所示。

```
from tkinter import *                                  #导入 Tkinter 库
def printkey(event):                                   #定义的函数监听键盘事件
    print('你按下了：'+event.char)
root=Tk()                                              #实例化 tk
entry=Entry(root)                                      #实例化一个单行输入框
#给输入框绑定按键监听事件<KeyPress>为监听任何按键
#<KeyPress-x>为监听某键 x，例如<KeyPress-A>为监听 A、<KeyPress-Return>为监听回车
entry.bind('<KeyPress>', printkey)
entry.pack()
root.mainloop()                                        #显示窗体
```

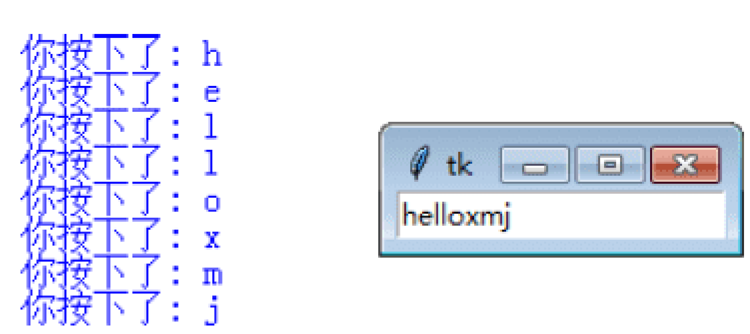


图 1-19　keyPress 键盘事件运行效果

例 **1-28** 获取鼠标单击标签时坐标的鼠标事件例子，运行效果如图 1-20 所示。

```
from tkinter import *                    #导入 Tkinter 库
def leftClick(event):                    #定义的函数监听鼠标事件
   print("x 轴坐标:", event.x)
   print("y 轴坐标:", event.y)
   print("相对于屏幕左上角 x 轴坐标:", event.x root)
   print("相对于屏幕左上角 y 轴坐标:", event.y root)
root=Tk()                                #实例化 tk
lab=Label(root,text="hello")             #实例化一个 Label
lab.pack()                               #显示 Label 组件
#给 Label 绑定鼠标监听事件
lab.bind("<Button-1>",leftClick)
root.mainloop()                          #显示窗体
```

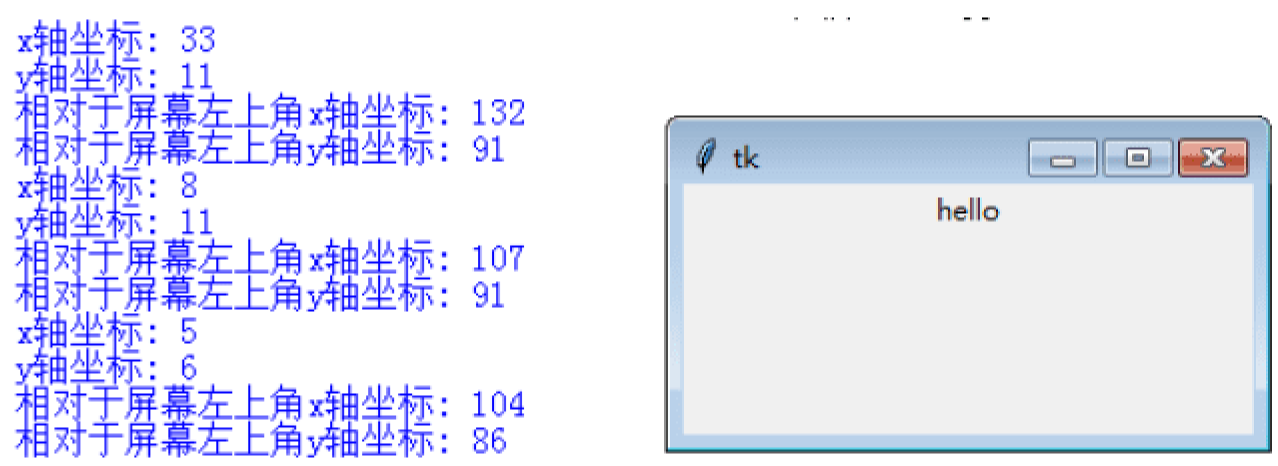


图 1-20 鼠标事件运行效果

1.4.6 图形界面设计应用案例——开发猜数字游戏

【案例】使用 Tkinter 开发猜数字游戏，运行效果如图 1-21 所示。

在该游戏中，计算机随机生成 1024 以内的数字，玩家去猜，猜的数字过大、过小都会给出提示，程序要统计玩家猜的次数。

```
import tkinter as tk
import random
number=random.randint(0,1024)              #玩家要猜的数字
running=True
num=0                                      #猜的次数
nmaxn=1024                                 #提示猜测范围的最大数
nminn=0                                    #提示猜测范围的最小数

def eBtnClose(event):                      #“关闭”按钮事件函数
    root.destroy()
def eBtnGuess(event):                      #“猜”按钮事件函数
    global nmaxn                           #全局变量
    global nminn
    global num
    global running
```

```
    if running:
        val_a=int(entry_a.get())     #获取猜的数字并转换成数字
        if val_a==number:
            labelqval("恭喜答对了！")
            num+=1
            running=False
            numGuess()                   #显示猜的次数
        elif val_a<number:               #猜小了
            if val_a>nminn:
                nminn=val_a              #修改提示猜测范围的最小数
                num+=1
                labelqval("小了哦,请输入"+str(nminn)+"到"+str(nmaxn)+"之间任意
                整数：")
        else:
            if val_a<nmaxn:
                nmaxn=val_a              #修改提示猜测范围的最大数
                num+=1
                labelqval("大了哦,请输入"+str(nminn)+"到"+str(nmaxn)+"之间任意
                整数：")
    else:
        labelqval('你已经答对啦...')
#显示猜的次数
def numGuess():
    if num==1:
        labelqval('一次答对！')
    elif num<10:
        labelqval('==十次以内就答对了牛。。。尝试次数：'+str(num))
    else:
        labelqval('好吧，您都试了超过10次了。。。。尝试次数：'+str(num))

def labelqval(vText):
    label_val_q.config(label_val_q,text=vText)      #修改提示标签文字

root=tk.Tk(className="猜数字游戏")
root.geometry("400x90+200+200")
label_val_q=tk.Label(root,width="80")                #提示标签
label_val_q.pack(side="top")

entry_a=tk.Entry(root,width="40")                    #单行输入文本框
btnGuess=tk.Button(root,text="猜")                   #“猜”按钮
entry_a.pack(side="left")
entry_a.bind('<Return>',eBtnGuess)                   #绑定事件
btnGuess.bind('<Button-1>',eBtnGuess)                #“猜”按钮
btnGuess.pack(side="left")
```

```
btnClose=tk.Button(root,text="关闭")                    #“关闭”按钮
btnClose.bind('<Button-1>',eBtnClose)
btnClose.pack(side="left")
labelqval("请输入 0 到 1024 之间任意整数：")
entry_a.focus_set()
print(number)
root.mainloop()
```

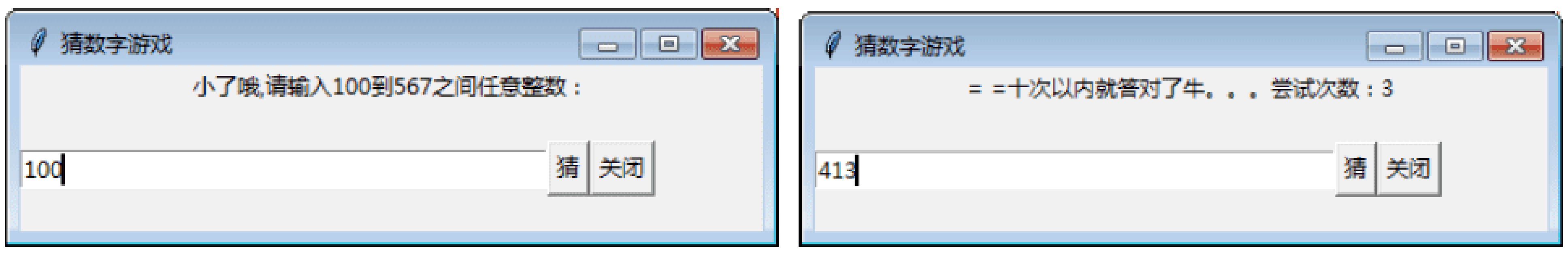


图 1-21　猜数字游戏运行效果

1.5　Python 文件的使用

在程序运行时，数据保存在内存的变量里。内存中的数据在程序结束或关机后就会消失。如果想要在下次开机运行程序时还使用同样的数据，就需要把数据存储在不易失的存储介质中，例如硬盘、光盘或 U 盘里，不易失存储介质上的数据保存在以存储路径命名的文件中。通过读/写文件，程序就可以在运行时保存数据。本节学习使用 Python 在磁盘上创建、读/写以及关闭文件。

视频讲解

使用文件跟人们在平时生活中使用记事本很相似。在使用记事本时需要先打开本子，使用之后要合上它。在打开记事本后，既可以读取信息，也可以向本子里写。不管是哪种情况，都需要知道在哪里进行读/写。在记事本中既可以一页一页从头到尾地读，也可以直接跳转到需要的地方。

在 Python 中对文件的操作通常按照以下 3 个步骤进行：

（1）使用 open()函数打开（或建立）文件，返回一个 file 对象。

（2）使用 file 对象的读/写方法对文件进行读/写操作。其中，将数据从外存传输到内存的过程称为读操作，将数据从内存传输到外存的过程称为写操作。

（3）使用 file 对象的 close()方法关闭文件。

视频讲解

1.5.1　打开/建立文件

在 Python 中要访问文件，必须打开 Python Shell 与磁盘上文件之间的连接。在使用 open()函数打开或建立文件时会建立文件和使用它的程序之间的连接，并返回代表连接的文件对象，通过文件对象就可以在文件所在的磁盘和程序之间传递文件内容，执行文件上的所有后续操作。文件对象有时也称为文件描述符或文件流。

在建立了 Python 程序和文件之间的连接后就创建了“流”数据，如图 1-22 所示。通常，程序使用输入流读出数据，使用输出流写入数据，就好像数据流入到程序并从程序中流出。在打开文件后才能读或写（或读并且写）文件内容。

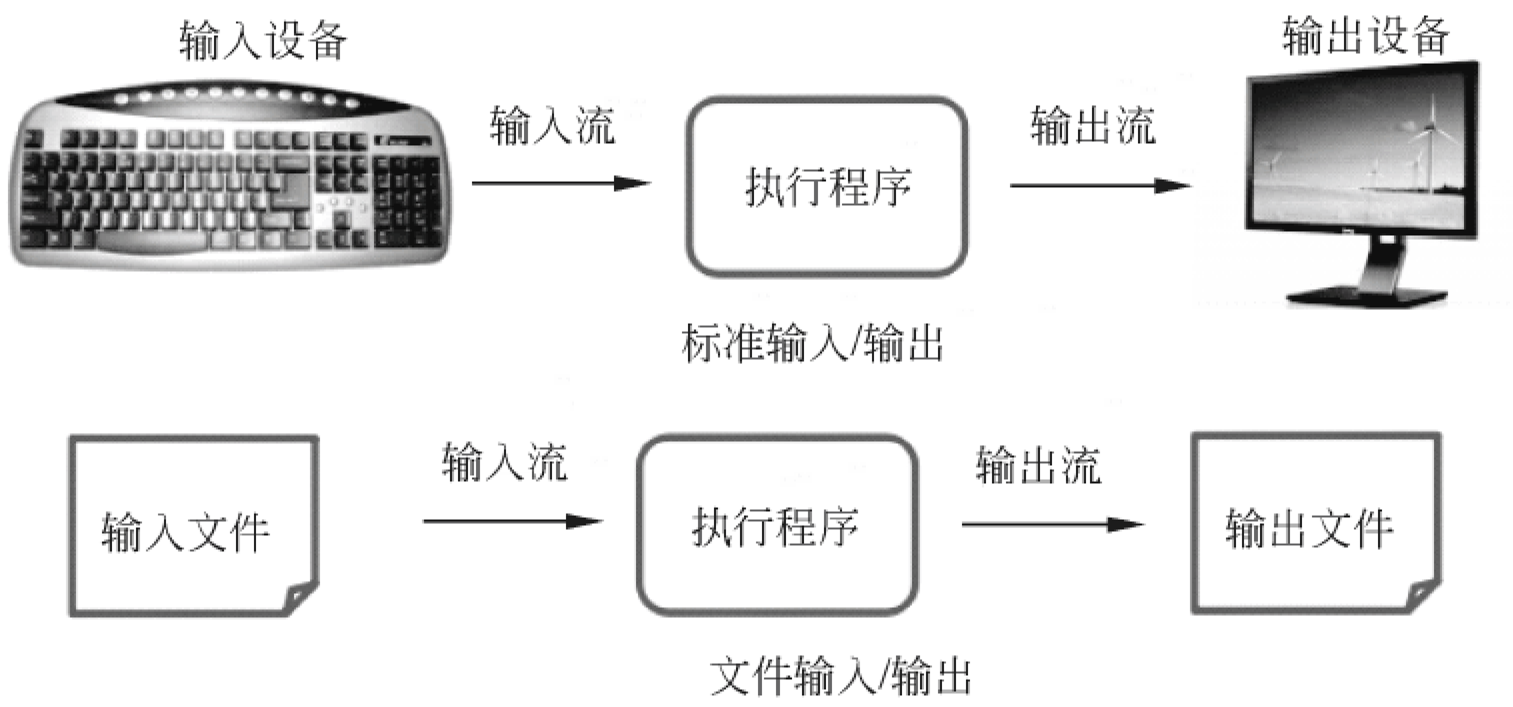


图 1-22　输入/输出流

open()函数用来打开文件。open()函数需要一个字符串路径，表明希望打开文件，并返回一个文件对象。其语法格式如下：

```
fileobj=open(filename[,mode[,buffering]])
```

其中，fileobj 是 open()函数返回的文件对象。参数 filename（文件名）是必写参数，它既可以是绝对路径，也可以是相对路径。模式（mode）和缓冲（buffering）为可选项。mode 是指明文件类型和操作的字符串，可以使用的值如表 1-12 所示。

表 1-12　open()函数中 mode 参数的常用值

值	描　述
'r'	读模式，如果文件不存在，则发生异常
'w'	写模式，如果文件不存在，则先创建文件再打开；如果文件存在，则清空文件内容再打开
'a'	追加模式，如果文件不存在，则先创建文件再打开；如果文件存在，打开文件后将新内容追加到原内容之后
'b'	二进制模式，可添加到其他模式中使用
'+'	读/写模式，可添加到其他模式中使用

说明：

（1）当 mode 参数省略时，可以获得能读取文件内容的文件对象，即'r'是 mode 参数的默认值。

（2）'+'参数指明读和写都是允许的，可以用到其他任何模式中。例如，可以用' r+'打开一个文本文件并读/写。

（3）'b'参数改变处理文件的方法。通常，Python 处理的是文本文件，当处理二进制文件（例如声音文件或图像文件）时应该在模式参数中增加'b'。例如，可以用'rb'来读取一个二进制文件。

open()函数的第 3 个参数——buffering 用来控制缓冲。当该参数取 0 或 False 时，I/O 是无缓冲的，所有读/写操作直接针对硬盘。当该参数取 1 或 True 时，I/O 有缓冲，此时

Python 使用内存代替硬盘，使程序的运行速度更快，只有在使用 flush()或 close()时才会将数据写入硬盘。当参数大于 1 时，表示缓冲区的大小，以字节为单位；负数表示使用默认缓冲区大小。

下面举例说明 open()函数的使用。

先用记事本创建一个文本文件，取名为 hello.txt，然后输入以下内容，保存在 D 盘的 python 文件夹中：

```
Hello!
Henan  Zhengzhou
```

在交互式环境中输入以下代码：

```
>>> helloFile=open("D:\\python\\hello.txt")
```

这条命令将以读取文本文件的方式打开放在 D 盘 Python 文件夹下的 hello.txt 文件。“读模式”是 Python 打开文件的默认模式。当文件以读模式打开时，只能从文件中读取数据，不能向文件写入或修改数据。

当调用 open()函数时将返回一个文件对象，在本例中文件对象保存在 helloFile 变量中。

```
>>> print(helloFile)
<_io.TextIOWrapper name='D:\\python\\hello.txt' mode='r' encoding='cp936'>
```

在打印文件对象时可以看到文件名、读/写模式和编码格式。cp936 是指 Windows 系统中的第 936 号编码格式，即 GB2312 的编码。接下来就可以调用 helloFile 文件对象的方法读取文件中的数据了。

1.5.2　读取文本文件

用户可以调用文件对象的多种方法读取文件内容。

❶ read()方法

不设置参数的 read()方法将整个文件的内容读取为一个字符串。read()方法一次读取文件的全部内容，性能根据文件大小而变化，例如 1GB 的文件在读取时需要使用同样大小的内存。

例 1-29　调用 read()方法读取 hello.txt 文件中的内容。

```
helloFile=open("D:\\python\\hello.txt")
fileContent=helloFile.read()
helloFile.close()
print(fileContent)
```

输出结果：

```
Hello!
Henan  Zhengzhou
```

用户也可以设置最大读入字符数来限制 read()函数一次返回的大小。

例 1-30 设置参数一次读取 3 个字符来读取文件。

```
helloFile=open("D:\\python\\hello.txt")
fileContent=""
whileTrue:
    fragment=helloFile.read(3)
    if fragment=="":    #或者 if not fragment
        break
    fileContent+=fragment
helloFile.close()
print(fileContent)
```

在读到文件结尾之后，read()方法会返回空字符串，此时 fragment==""成立，退出循环。

❷ readline()方法

readline()方法从文件中获取一个字符串，每个字符串就是文件中的每一行。

例 1-31 调用 readline()方法读取 hello.txt 文件中的内容。

```
helloFile=open("D:\\python\\hello.txt")
fileContent=""
whileTrue:
   line=helloFile.readline()
   if line=="":    #或者 if not line
      break
   fileContent+=line
helloFile.close()
print(fileContent)
```

当读取到文件结尾之后，readline()方法同样返回空字符串，使得 line==""成立，跳出循环。

❸ readlines()方法

readlines()方法返回一个字符串列表，其中的每一项是文件中每一行的字符串。

例 1-32 使用 readlines()方法读取文件内容。

```
helloFile=open("D:\\python\\hello.txt")
fileContent=helloFile.readlines()
helloFile.close()
print(fileContent)
for line in fileContent:   #输出列表
   print(line)
```

readlines()方法也可以设置参数，指定一次读取的字符数。

1.5.3 写文本文件

视频讲解

写文件和读文件相似，都需要先创建文件对象连接，所不同的是，打

开文件时是以写模式或添加模式打开。如果文件不存在，则创建该文件。

与读文件时不能添加或修改数据类似，写文件时也不允许读取数据。在用写模式打开已有文件时会覆盖文件的原有内容，从头开始，就像用一个新值覆写一个变量的值。

```
>>> helloFile=open("D:\\python\\hello.txt","w")
                                  #用写模式打开已有文件时会覆盖文件的原有内容
>>> fileContent=helloFile.read()
Traceback(most recent call last):
  File "<pyshell#1>", line 1, in <module>
    fileContent=helloFile.read()
IOError: File not open for reading
>>> helloFile.close()
>>> helloFile=open("D:\\python\\hello.txt")
>>> fileContent=helloFile.read()
>>> len(fileContent)
0
>>> helloFile.close()
```

由于用写模式打开已有文件时文件的原有内容被清空，所以再次读取内容时长度为 0。

❶ write()方法

write()方法用于将字符串参数写入文件。

例 1-33　用 write()方法写文件。

```
helloFile=open("D:\\python\\hello.txt","w")
helloFile.write("First line.\nSecond line.\n")
helloFile.close()
helloFile=open("D:\\python\\hello.txt","a")
helloFile.write("third line. ")
helloFile.close()
helloFile=open("D:\\python\\hello.txt")
fileContent=helloFile.read()
helloFile.close()
print(fileContent)
```

运行结果：

```
First line.
Second line.
third line.
```

当以写模式打开文件 hello.txt 时，文件的原有内容被覆盖。调用 write()方法将字符串参数写入文件，这里“\n”代表换行符。关闭文件之后再次以添加模式打开文件 hello.txt，调用 write()方法写入的字符串“third line.”被添加到了文件末尾。最终以读模式打开文件后读取到的内容共有 3 行字符串。

注意，write()方法不能自动在字符串末尾添加换行符，需要用户自己添加“\n”。

例 1-34　完成一个自定义函数 copy_file()，实现文件的复制功能。

copy_file()函数需要两个参数，指定需要复制的文件 oldfile 和文件的备份 newfile。分别以读模式和写模式打开两个文件，从 oldfile 一次读入 50 个字符并写入 newfile。当读到

文件末尾时 fileContent==""成立，退出循环并关闭两个文件。

```
def copy file(oldfile,newfile):
    oldFile=open(oldfile,"r")
    newFile=open(newfile,"w")
    whileTrue:
        fileContent=oldFile.read(50)
        if fileContent=="":   #读到文件末尾时
            break
        newFile.write(fileContent)
    oldFile.close()
    newFile.close()
    return
copy_file("D:\\python\\hello.txt","D:\\python\\hello2.txt")
```

❷ writelines()方法

writelines(sequence)方法向文件写入一个序列字符串列表，如果需要换行则要自己加入每行的换行符。

```
obj=open("log.txt","w")
list2=["11", "test", "hello", "44", "55"]
obj.writelines(list2)
obj.close()
```

运行结果是生成一个"log.txt"文件，内容是“11testhello4455”，可见没有换行。另外注意 writelines()方法写入的序列必须是字符串序列，整数序列会产生错误。

1.5.4 文件内移动

视频讲解

无论是读或写文件，Python 都会跟踪文件中的读/写位置。在默认情况下，文件的读/写都是从文件的开始位置进行。Python 提供了控制文件读/写起始位置的方法，使得用户可以改变文件读/写操作发生的位置。

当使用 open()函数打开文件时，open()函数在内存中创建缓冲区，将磁盘上的文件内容复制到缓冲区。在文件内容复制到文件对象缓冲区后，文件对象将缓冲区视为一个大的列表，其中的每一个元素都有自己的索引，文件对象按字节对缓冲区索引计数。同时，文件对象对文件当前位置（即当前读/写操作发生的位置）进行维护，如图 1-23 所示。许多方法隐式使用当前位置。例如在调用 readline()方法后，文件当前位置移动到下一个回车处。

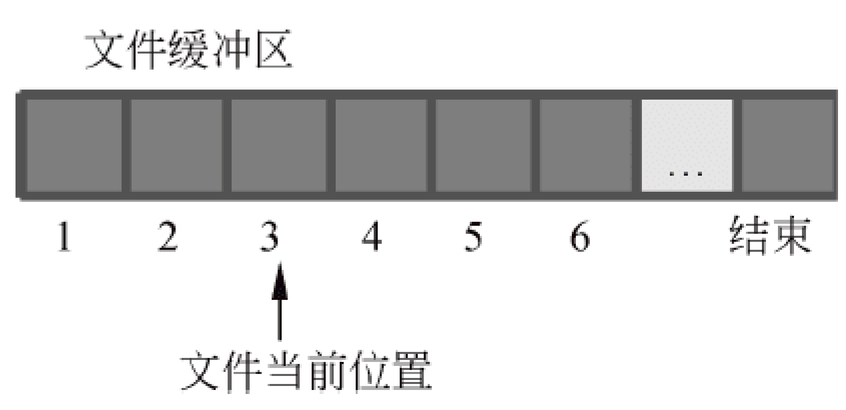


图 1-23　文件当前位置

Python 使用一些函数跟踪文件当前位置。tell()函数可以计算文件当前位置和开始位置之间的字节偏移量。

```
>>> exampleFile=open("D:\\python\\example.txt","w")
>>> exampleFile.write("0123456789")
>>> exampleFile.close()
>>> exampleFile=open("D:\\python\\example.txt")
>>> exampleFile.read(2)
'01'
>>> exampleFile.read(2)
'23'
>>> exampleFile.tell()
4
>>> exampleFile.close()
```

这里，exampleFile.tell()函数返回的是一个整数 4，表示文件当前位置和开始位置之间有 4 个字节偏移量。因为已经从文件中读取 4 个字符了，所以为 4 个字节偏移量。

seek()函数设置新的文件当前位置，允许在文件中跳转，实现对文件的随机访问。

seek()函数有两个参数，第 1 个参数是字节数，第 2 个参数是引用点。seek()函数将文件当前指针由引用点移动指定的字节数到指定的位置。其语法格式如下：

```
seek(offset[,whence])
```

说明：offset 是一个字节数，表示偏移量；引用点 whence 有下面 3 个取值。

- 文件开始处为 0，这也是默认取值，意味着使用该文件的开始处作为基准位置，此时字节偏移量必须非负。
- 当前文件位置为 1，则是使用当前位置作为基准位置，此时偏移量可以取负值。
- 文件结尾处为 2，则该文件的末尾将被作为基准位置。

1.5.5　文件的关闭

用户应该牢记使用 close()方法关闭文件。关闭文件是取消程序和文件之间连接的过程，内存缓冲区中的所有内容将写入磁盘，因此必须在使用文件后关闭文件确保信息不会丢失。

如果要确保文件关闭，可以使用 try/finally 语句，在 finally 子句中调用 close()方法：

```
helloFile=open("D:\\python\\hello.txt","w")
try :
    helloFile.write("Hello,Sunny Day!")
finally:
    helloFile.close()
```

也可以使用 with 语句自动关闭文件：

```
with open("D:\\python\\hello.txt") as helloFile:
    s=helloFile.read()
print(s)
```

with 语句可以打开文件并赋值给文件对象，之后就可以对文件进行操作了。文件会在语句结束后自动关闭，即使是由于异常引起的结束也是如此。

1.5.6　二进制文件的读/写

Python 没有二进制类型，但是可以用 string（字符串）类型来存储二进制类型数据，因为 string 是以字节为单位的。

视频讲解

❶ 数据转换成字节串

pack()方法可以把数据转换成字节串（以字节为单位的字符串）。

格式如下：

```
pack(格式化字符串, 数据)
```

格式化字符串中可用的格式字符见表 1-13 中的格式字符。例如：

```
import struct
a=20
bytes=struct.pack('i',a)            #将 a 变为字符串
print(bytes)
```

结果如下：

```
b'\x14\x00\x00\x00'
```

此时 bytes 就是一个字符串，该字符串按字节和 a 的二进制存储内容相同。在结果中\x 是十六进制的意思，20 的十六进制是 14。

如果是由多个数据构成的，可以这样：

```
a='hello'
b='world!'
c=2
d=45.123
bytes=struct.pack('5s6sif',a.encode('utf-8'),b.encode('utf-8'),c,d)
```

'5s6sif'就是格式化字符串，由数字加字符构成。其中，5s 表示占 5 个字符宽度的字符串，2i 表示两个整数。表 1-13 是可用的格式字符及对应 C 语言、Python 语言中的类型。

表 1-13　格式字符

格 式 字 符	C 语言中的类型	Python 语言中的类型	字 节 数
c	char	string of length 1	1
b	signed char	integer	1
B	unsigned char	integer	1
?	_Bool	bool	1
h	short	integer	2
H	unsigned short	integer	2
i	int	integer	4
I	unsigned int	integer or long	4

续表

格 式 字 符	C 语言中的类型	Python 语言中的类型	字　节　数
l	long	integer	4
L	unsigned long	long	4
q	long long	long	8
Q	unsigned long long	long	8
f	float	float	4
d	double	float	8
s	char[]	string	1
p	char[]	string	1
P	void *	long	与 OS 有关

```
bytes=struct.pack('5s6sif',a.encode('utf-8'),b.encode('utf-8'),c,d)
```

此时的 bytes 就是二进制形式的数据了，可以直接写入文件。例如：

```
binfile=open("D:\\python\\hellobin.txt","wb")
binfile.write(bytes)
binfile.close()
```

❷ 字节串还原成数据

unpack()方法可以把相应数据的字节串（以字节为单位的字符串）还原成数据。

```
bytes=struct.pack('i',20)              #将 20 变为字符串
```

再进行反操作，现有二进制数据 bytes（其实就是字符串），将它反过来转换成 Python 的数据类型：

```
a,=struct.unpack('i',bytes)
```

注意，unpack()返回的是元组。所以，如果只有一个变量：

```
bytes=struct.pack('i',a)
```

那么解码的时候需要这样：

```
a,=struct.unpack('i',bytes) 或者(a,)=struct.unpack('i',bytes)
```

如果直接用 a=struct.unpack('i',bytes)，那么 a=(20,)，是一个 tuple 而不是原来的整数。

例如把 D 盘 python 文件夹下 hellobin.txt 文件中的数据读取并显示。

```
import struct
binfile=open("D:\\python\\hellobin.txt","rb")
bytes=binfile.read()
(a,b,c,d)=struct.unpack('5s6sif',bytes)
    #通过 struct.unpack()解码成 Python 变量
t=struct.unpack('5s6sif',bytes)
    #通过 struct.unpack()解码成元组
print(t)
```

读取结果：

```
(b'hello', b'world!', 2, 45.12300109863281)
```

1.6 Python 的第三方库

Python 语言有标准库和第三方库两类库，标准库随 Python 安装包一起发布，用户可以随时使用，第三方库需要在安装后才能使用。由于 Python 语言经历了数次版本更迭，而且第三方库由全球开发者分布式维护，缺少统一的集中管理，所以 Python 第三方库曾经一度制约了 Python 语言的普及和发展。随着官方 pip 工具的应用，Python 第三方库的安装变得十分容易。常用 Python 第三方库如表 1-14。

表 1-14 常用 Python 第三方库

库 名 称	库 用 途
Django	开源 Web 开发框架，它鼓励快速开发，并遵循 MVC 设计，比较好用，开发周期短
webpy	一个小巧灵活的 Web 框架，虽然简单，但是功能强大
Matplotlib	用 Python 实现的类 matlab 的第三方库，用于绘制一些高质量的数学二维图形
SciPy	基于 Python 的 matlab 实现，旨在实现 matlab 的所有功能
NumPy	基于 Python 的科学计算第三方库，提供了矩阵、线性代数、傅立叶变换等解决方案
PyGtk	基于 Python 的 GUI 程序开发 GTK+库
PyQt	用于 Python 的 QT 开发库
WxPython	Python 下的 GUI 编程框架，与 MFC 的架构相似
BeautifulSoup	基于 Python 的 HTML/XML 解析器，简单易用
PIL	基于 Python 的图像处理库，功能强大，对图形文件的格式支持广泛
MySQLdb	用于连接 MySQL 数据库
Pygame	基于 Python 的多媒体开发和游戏软件开发模块
Py2exe	将 Python 脚本转换为 Windows 上可以独立运行的可执行程序
pefile	Windows PE 文件解析器

最常用且最高效的 Python 第三方库的安装方式是采用 pip 工具安装。pip 是 Python 官方提供并维护的在线第三方库安装工具。对于同时安装 Python2 和 Python3 环境为系统的情况，建议采用 pip3 命令专门为 Python3 版安装第三方库。

例如安装 Pygame 库，pip 工具默认从网络上下载 Pygame 库安装文件并自动装到系统中。注意，pip 是在命令行下（cmd）运行的工具。

```
D:\>pip install pygame
```

用户也可以卸载 Pygame 库，卸载过程可能需要用户确认。

```
D:\>pip uninstall pygame
```

用户还可以通过 list 子命令列出当前系统中已经安装的第三方库，例如：

```
D:\>pip list
```

pip 是 Python 第三方库最主要的安装方式，可以安装超过 90%以上的第三方库。然而，由于一些历史、技术等原因，还有一些第三方库暂时无法用 pip 安装，此时需要用其他的安装方法（例如下载库文件后手工安装）。

第 2 章

序列应用——猜单词游戏

2.1 猜单词游戏功能介绍

视频讲解

猜单词游戏就是计算机随机产生一个单词，打乱字母顺序，供玩家去猜。此游戏采用控制字符界面，运行界面如图 2-1 所示。

```
    欢迎参加猜单词游戏
   把字母组合成一个正确的单词.

乱序后单词: luebjm

请你猜: jumble
真棒, 你猜对了!
是否继续 (Y/N): y
乱序后单词: oionispt

请你猜: position
真棒, 你猜对了!
是否继续 (Y/N): y
乱序后单词: tsinoiop

请你猜:
```

图 2-1　猜单词游戏程序的运行界面

2.2 程序设计的思路

在游戏中需要随机产生单词以及数字，所以引入 random 模块随机数函数，其中 random.choice()可以从序列中随机选取元素。例如：

```
#创建单词序列元组
WORDS=("python", "jumble", "easy", "difficult", "answer", "continue",
       "phone", "position", "position", "game")
```

```
#从序列中随机挑出一个单词
word=random.choice(WORDS)
```

word 就是从单词序列中随机挑出一个单词。

在游戏中随机挑出一个单词 word 后，如何把单词 word 的字母顺序打乱？方法是随机从单词字符串中选择一个位置 position，把 position 位置的那个字母加入乱序后单词 jumble，同时将原单词 word 中 position 位置的那个字母删去（通过连接 position 位置前字符串和其后字符串实现）。通过多次循环就可以产生新的乱序后单词 jumble。

```
while word:          #word 不是空串循环
    #根据 word 长度产生 word 的随机位置
    position=random.randrange(len(word))
    #将 position 位置的字母组合到乱序后单词
    jumble+=word[position]
    #通过切片，将 position 位置的字母从原单词中删除
    word=word[:position]+word[(position+1):]
print("乱序后单词:", jumble)
```

2.3 关键技术——random 模块

random 模块可以产生一个随机数或者从序列中获取一个随机元素，它的常用方法和使用例子如下：

❶ 常用方法

1）random.random()

random.random()用于生成一个随机小数 n，$0 \le n < 1.0$。

```
import random
random.random()
```

执行以上代码的输出结果如下：

```
0.85415370477785668
```

2）random.uniform(a, b)

random.uniform(a, b)，用于生成一个指定范围内的随机小数，两个参数中一个是上限，一个是下限。如果 $a<b$，则生成的随机数 n 有 $a \le n \le b$；如果 $a>b$，则 $b \le n \le a$。

代码如下：

```
import random
print(random.uniform(10, 20))
print(random.uniform(20, 10))
```

执行以上代码的输出结果如下：

```
14.247256006293084
15.53810495673216
```

3）random.randint(a,b)

random.randint(a,b)用于随机生成一个指定范围内的整数，其中参数 a 是下限，参数 b 是上限，生成的随机数 n 有 a≤n≤b。

```
import random
print(random.randint(12, 20))   #生成的随机数 n 有 12≤n≤20
print(random.randint(20, 20))   #结果永远是 20
#print(random.randint(20, 10)) #该语句是错误的，下限必须小于上限
```

4）random.randrange([start],stop[,step])

random.randrange([start],stop[,step])用于从指定范围内按指定基数递增的集合中获取一个随机数。例如 random.randrange(10,100,2)，结果相当于从[10,12,14,16,…,96,98]序列中获取一个随机数。random.randrange(10,100,2)在结果上与 random.choice(range(10,100,2))等效。

5）random.choice()

random.choice()从序列中获取一个随机元素，其原型为 random.choice(sequence)，参数 sequence 表示一个有序类型。这里要说明一下，sequence 在 Python 中不是一种特定的类型，而是泛指序列数据结构。列表、元组、字符串都属于 sequence。下面是使用 random.choice()的一些例子：

```
import random
print(random.choice("学习 Python"))                    #从字符串中随机取一个字符
print(random.choice(["JGood", "is", "a", "handsome", "boy"]))
                                                        #从 list 列表中随机取
print(random.choice(("Tuple", "List", "Dict")))        #从 tuple 元组中随机取
```

执行以上代码的输出结果如下：

```
学
is
Dict
```

当然，每次运行的结果都不一样。

6）random.shuffle(x[,random])

random.shuffle(x[, random])用于将一个列表中的元素打乱，例如：

```
p=["Python", "is", "powerful", "simple", "and so on..."]
random.shuffle(p)
print(p)
```

执行以上代码的输出结果如下：

```
['powerful', 'simple', 'is', 'Python', 'and so on...']
```

在发牌游戏案例中使用此方法打乱牌的顺序实现洗牌功能。

7）random.sample(sequence, k)

random.sample(sequence, k)用于从指定序列中随机获取指定长度的片段，sample()函数

不会修改原有序列。

```
list=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
slice=random.sample(list, 5)        #从 list 中随机获取 5 个元素作为一个片段返回
print(slice)
print(list)                         #原有序列并没有改变
```

执行以上代码的输出结果如下：

```
[5, 2, 4, 9, 7]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

❷ 使用举例

以下是常用情况举例：

1）随机字符

```
>>> import random
>>> random.choice('abcdefg&#%^*f')
```

其结果为'd'。

2）在多个字符中选取特定数量的字符

```
>>> import random
>>>random.sample('abcdefghij', 3)
```

其结果为['a', 'd', 'b']。

3）在多个字符中选取特定数量的字符组成新字符串

```
>>> import random
>>> " ".join(random.sample(['a','b','c','d','e','f','g','h','i','j'],
3)).replace(" ","")
```

其结果为'ajh'。

4）随机选取字符串

```
>>> import random
>>> random.choice(['apple', 'pear', 'peach', 'orange', 'lemon'])
```

其结果为'lemon'。

5）洗牌

```
>>> import random
>>> items=[1, 2, 3, 4, 5, 6]
>>> random.shuffle(items)
>>> items
```

其结果为 [3,2,5,6,4,1]。

6）随机选取 0 到 100 之间的偶数

```
>>> import random
>>> random.randrange(0, 101, 2)
```

其结果为 42。

7）随机生成 1～100 之间的小数

```
>>> random.uniform(1, 100)
```

其结果为 5.4221167969800881。

2.4　程序设计的步骤

猜单词游戏程序的代码如下：

```
#猜单词游戏
import random
#创建单词序列
WORDS=("python", "jumble", "easy", "difficult", "answer", "continue",
       "phone", "position", "pose", "game")
#开始游戏
print(
"""
     欢迎参加猜单词游戏
   把字母组合成一个正确的单词.
"""
)
iscontinue="y"
while iscontinue=="y" or iscontinue=="Y":
    #从序列中随机挑出一个单词
    word=random.choice(WORDS)
    #一个用于判断玩家是否猜对的变量
    correct=word
    #创建乱序后单词
    jumble=""
    while word:      #word 不是空串循环
        #根据 word 长度产生 word 的随机位置
        position=random.randrange(len(word))
        #将 position 位置的字母组合到乱序后单词
        jumble+=word[position]
        #通过切片将 position 位置的字母从原单词中删除
        word=word[:position]+word[(position+1):]
    print("乱序后单词:", jumble)

    guess=input("\n 请你猜: ")
    while guess!=correct and guess!="":
        print("对不起不正确.")
        guess=input("继续猜: ")
```

```
    if guess==correct:
        print("真棒，你猜对了！")
  iscontinue=input("\n是否继续（Y/N）：")
```

运行结果：

```
     欢迎参加猜单词游戏
  把字母组合成一个正确的单词.
乱序后单词： yaes
请你猜： easy
真棒，你猜对了！
是否继续（Y/N）：y
乱序后单词： diufctlfi
请你猜： difficutl
对不起不正确.
继续猜： difficult
真棒，你猜对了！
是否继续（Y/N）：n
>>>
```

第 3 章

数据库应用——智力问答测试

3.1 智力问答测试功能介绍

智力问答测试，内容涉及历史、经济、风情、民俗、地理、人文等古今中外各方面的知识，让玩家在轻松娱乐、益智、搞笑的同时不知不觉地增长知识。在答题过程中对做对、做错进行实时跟踪，测试完成后能根据玩家的答题情况给出成绩。程序运行界面如图 3-1 所示。

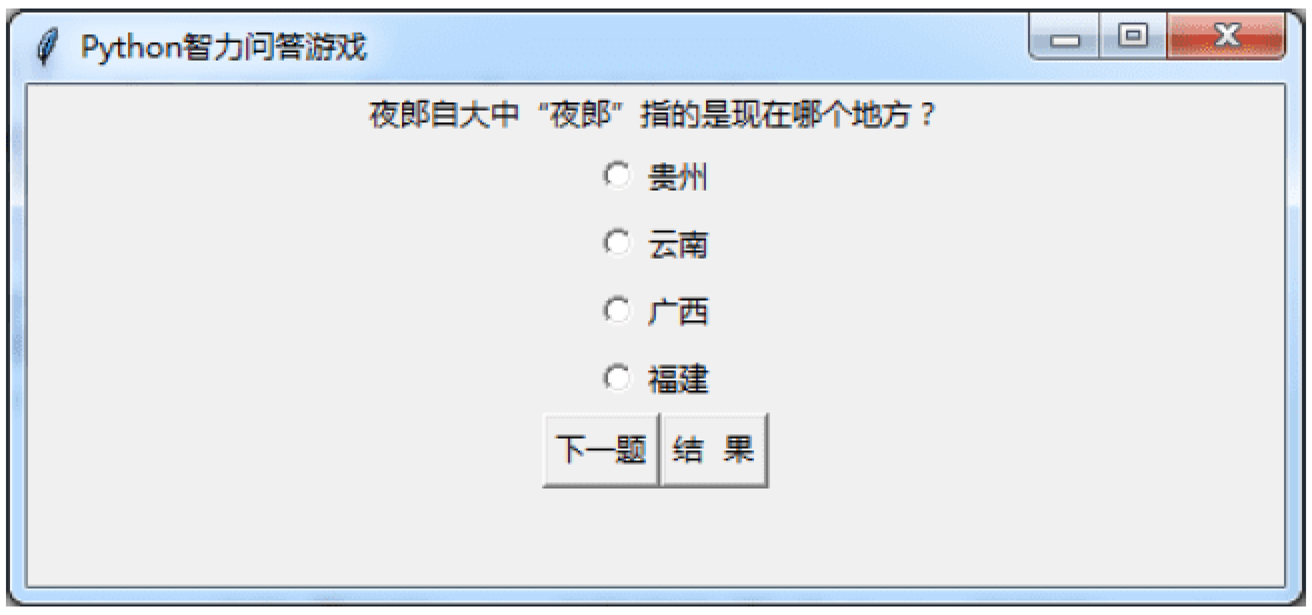


图 3-1　智力问答测试程序的运行界面

3.2 程序设计的思路

程序使用了一个 SQLite 试题库 test2.db，其中每个智力问答由题目、4 个选项和正确答案组成（question、Answer_A、Answer_B、Answer_C、Answer_D、right_Answer）。在测试前，程序从试题库 test2.db 读取试题信息，存储到 values 列表中。在测试时，顺序从 values 列表读出题目显示在 GUI 界面中供用户答题。在进行界面设计时，智力问答题目是标签控件，4 个选项是单选按钮控件，在“下一题”按钮单击事件中实现题目切换和对错判断，如果正确则得分 score 加 10 分，错误不加分，并判断用户是否做完。在“结果”按

钮单击事件中实现得分 score 的显示。

3.3 关键技术

从 Python2.5 版本以上就内置了 SQLite3，所以在 Python 中使用 SQLite 不需要安装任何东西，直接使用。SQLite3 数据库使用 SQL 语言。SQLite 作为后端数据库，可以制作有数据存储需求的工具。Python 标准库中的 SQLite3 提供该数据库的接口。

视频讲解

3.3.1 访问数据库的步骤

从 Python2.5 开始，SQLite3 就成了 Python 的标准模块，这也是 Python 中的唯一一个数据库接口类模块，大大方便了用户使用 Python SQLite 数据库开发小型数据库应用系统。

Python 的数据库模块有统一的接口标准，所以数据库操作有统一的模式，操作数据库 SQLite3 主要分为以下几步：

❶ 导入 Python SQLite 数据库模块

Python 标准库中带有 sqlite3 模块，可直接导入：

```
import sqlite3
```

❷ 建立数据库连接，返回 Connection 对象

使用数据库模块的 connect()函数建立数据库连接，返回连接对象 con。

```
con=sqlite3.connect(connectstring)  #连接到数据库，返回 sqlite3.connection 对象
```

说明：connectstring 是连接字符串。对于不同的数据库连接对象，其连接字符串的格式不同，sqlite 的连接字符串为数据库的文件名，例如“E:\test.db”。如果指定连接字符串为 memory，则可创建一个内存数据库。例如：

```
import sqlite3
con=sqlite3.connect("E:\\test.db")
```

如果 E 盘下的 test.db 存在，则打开数据库；否则在该路径下创建数据库 test.db 并打开。

❸ 创建游标对象

使用游标对象能够灵活地对从表中检索出的数据进行操作，就本质而言，游标实际上是一种能从包括多条数据记录的结果集中每次提取一条记录的机制。

调用 con.cursor()创建游标对象 cur：

```
cur=con.cursor()            #创建游标对象
```

❹ 使用 Cursor 对象的 execute()方法执行 SQL 命令返回结果集

调用 cur.execute()、cur.executemany()、cur.executescript()方法查询数据库。

- cur.execute(sql)：执行 SQL 语句。
- cur.execute(sql, parameters)：执行带参数的 SQL 语句。

- cur.executemany(sql, seq_of_pqrameters)：根据参数执行多次 SQL 语句。
- cur.executescript(sql_script)：执行 SQL 脚本。

例如创建一个表 category。

```
cur.execute("create table category(id primary key, sort, name)")
```

此时将创建一个包含 3 个字段 id、sort 和 name 的表 category。下面向表中插入记录：

```
cur.execute("insert into category values(1, 1, 'computer')")
```

在 SQL 语句字符串中可以使用占位符“?”表示参数，传递的参数使用元组。例如：

```
cur.execute("insert into category values (? , ? ,? ) ", (2, 3, 'literature'))
```

❺ 获取游标的查询结果集

调用 cur.fetchall()、cur.fetchone()、cur.fetchmany()返回查询结果。

- cur.fetchone()：返回结果集的下一行（Row 对象）；无数据时返回 None。
- cur.fetchall()：返回结果集的剩余行（Row 对象列表），无数据时返回空 List。
- cur.fetchmany()：返回结果集的多行（Row 对象列表），无数据时返回空 List。

例如：

```
cur.execute("select * from catagory")
print(cur.fetchall())      #提取查询到的数据
```

返回结果如下：

```
[(1, 1, 'computer'), (2, 2, 'literature')]
```

如果使用 cur.fetchone()，则首先返回列表中的第 1 项，再次使用，返回第 2 项，依次进行。

用户也可以直接使用循环输出结果，例如：

```
for row in cur.execute("select * from catagory"):
   print(row[0],row[1])
```

❻ 数据库的提交和回滚

根据数据库事物隔离级别的不同，可以提交或回滚。

- con.commit()：事务提交。
- con.rollback()：事务回滚。

❼ 关闭 Cursor 对象和 Connection 对象

最后需要关闭打开的 Cursor 对象和 Connection 对象。

- cur.close()：关闭 Cursor 对象。
- con.close()：关闭 Connection 对象。

3.3.2　创建数据库和表

例 3-1　创建数据库 sales，并在其中创建表 book，表中包含 3 列，即 id、price 和 name，

其中 id 为主键（primary key）。

```
#导入 Python SQLite 数据库模块
import sqlite3
#创建 SQLite 数据库
con=sqlite3.connect("E:\sales.db")
#创建表 book，包含 3 列，即 id（主键）、price 和 name
con.execute("create table book(id primary key, price, name)")
```

说明：Connection 对象的 execute()方法是 Cursor 对象对应方法的快捷方式，系统会创建一个临时 Cursor 对象，然后调用对应的方法，并返回 Cursor 对象。

3.3.3 数据库的插入、更新和删除操作

在数据库表中插入、更新、删除记录的一般步骤如下：

（1）建立数据库连接。

（2）创建游标对象 cur，使用 cur.execute(sql)执行 SQL 的 insert、update、delete 等语句完成数据库记录的插入、更新、删除操作，并根据返回值判断操作结果。

（3）提交操作。

（4）关闭数据库。

例 3-2 数据库表记录的插入、更新和删除操作。

```
import sqlite3
books=[("021",25,"大学计算机"),("022",30, "大学英语"),("023",18, "艺术欣赏"),
( "024",35, "高级语言程序设计")]
#打开数据库
Con=sqlite3.connect("E:\sales.db")
#创建游标对象
Cur=Con.cursor()
#插入一行数据
Cur.execute("insert into book(id,price,name) values('001',33,'大学计算机
多媒体')")
Cur.execute("insert into book(id,price,name) values(?,?,?) " ,("002",28,
"数据库基础"))
#插入多行数据
Cur.executemany("insert into book(id,price,name) values (?,?,?) ",books)
#修改一行数据
Cur.execute("Update book set price=? where name=? ",(25,"大学英语"))
#删除一行数据
n=Cur.execute("delete from book where price=?",(25,))
print("删除了",n.rowcount,"行记录")
Con.commit()
Cur.close()
Con.close()
```

运行结果如下：

```
删除了 2 行记录
```

3.3.4　数据库表的查询操作

查询数据库的步骤如下：

（1）建立数据库连接。

（2）创建游标对象 cur，使用 cur.execute(sql)执行 SQL 的 select 语句。

（3）循环输出结果。

```
import sqlite3
#打开数据库
Con=sqlite3.connect("E:\sales.db")
#创建游标对象
Cur=Con.cursor()
#查询数据库表
Cur.execute("select id,price,name from book")
for row in Cur:
    print(row)
```

运行结果如下：

```
('001', 33, '大学计算机多媒体')
('002', 28, '数据库基础')
('023', 18, '艺术欣赏')
('024', 35, '高级语言程序设计')
```

3.3.5　数据库使用实例——学生通讯录

设计一个学生通讯录，可以添加、删除、修改里面的信息。

```
import sqlite3
#打开数据库
def opendb():
        conn=sqlite3.connect("E:\mydb.db")
        cur=conn.execute("create table if not exists tongxinlu(usernum
        integer primary key,username varchar(128), passworld varchar(128),
        address varchar(125), telnum varchar(128))")
        return cur, conn
#查询全部信息
def showalldb():
        print("-------------------处理后的数据-------------------")
        hel=opendb()
        cur=hel[1].cursor()
```

```
        cur.execute("select * from tongxinlu")
        res=cur.fetchall()
        for line in res:
            for h in line:
                print(h),
            print
        cur.close()
#输入信息
def into():
        usernum=input("请输入学号：")
        username1=input("请输入姓名：")
        passworld1=input("请输入密码：")
        address1=input("请输入地址：")
        telnum1=input("请输入联系电话：")
        return usernum,username1, passworld1, address1, telnum1
#往数据库中添加内容
def adddb():
        welcome="""----------------欢迎使用添加数据功能----------------"""
        print(welcome)
        person=into()
        hel=opendb()
        hel[1].execute("insert into tongxinlu(usernum,username, passworld,
        address, telnum)values(?,?,?,?,?)",(person[0], person[1],
        person[2], person[3],person[4]))
        hel[1].commit()
        print ("----------------恭喜你，数据添加成功----------------")
        showalldb()
        hel[1].close()
#删除数据库中的内容
def deldb():
        welcome="------------------欢迎使用删除数据库功能------------------"
        print(welcome)
        delchoice=input("请输入想要删除学号：")
        hel=opendb()                #返回游标 conn
        hel[1].execute("delete from tongxinlu where usernum ="+delchoice)
        hel[1].commit()
        print ("----------------恭喜你，数据删除成功----------------")
        showalldb()
        hel[1].close()
#修改数据库的内容
def alter():
        welcome="------------------欢迎使用修改数据库功能-----------------"
        print(welcome)
        changechoice=input("请输入想要修改的学生的学号：")
        hel=opendb()
```

```
        person=into()
        hel[1].execute("update tongxinlu set usernum=?,username=?,
        passworld=?,address=?,telnum=? where usernum="+changechoice,
        (person[0], person[1], person[2], person[3],person[4]))
        hel[1].commit()
        showalldb()
        hel[1].close()
#查询数据
def searchdb():
        welcome="------------------欢迎使用查询数据库功能-----------------"
        print(welcome)
        choice=input("请输入要查询的学生的学号：")
        hel=opendb()
        cur=hel[1].cursor()
        cur.execute("select * from tongxinlu where usernum="+choice)
        hel[1].commit()
              print("---------------恭喜你，你要查找的数据如下--------------")
        for row in cur:
                print(row[0],row[1],row[2],row[3],row[4])
             cur.close()
        hel[1].close()
#是否继续
def conti(a):
        choice=input("是否继续？（y or n）:")
        if choice=='y':
                a=1
        else:
                a=0
        return a
if __name__=="__main__":
        flag=1
        while flag:
                welcome="---------欢迎使用数据库通讯录---------"
                print(welcome)
                choiceshow="""
请选择您的进一步选择：
（添加）往数据库里面添加内容
（删除）删除数据库中内容
（修改）修改数据库的内容
（查询）查询数据库的内容
选择您想要进行的操作：
"""
                choice=input(choiceshow)
                if choice=="添加":
                        adddb()
```

```
            conti(flag)
    elif choice=="删除":
            deldb()
            conti(flag)
    elif choice=="修改":
            alter()
            conti(flag)
    elif choice=="查询":
            searchdb()
            conti(flag)
    else:
            print("你输入错误，请重新输入")
```

程序运行界面和添加记录界面如图 3-2 所示。

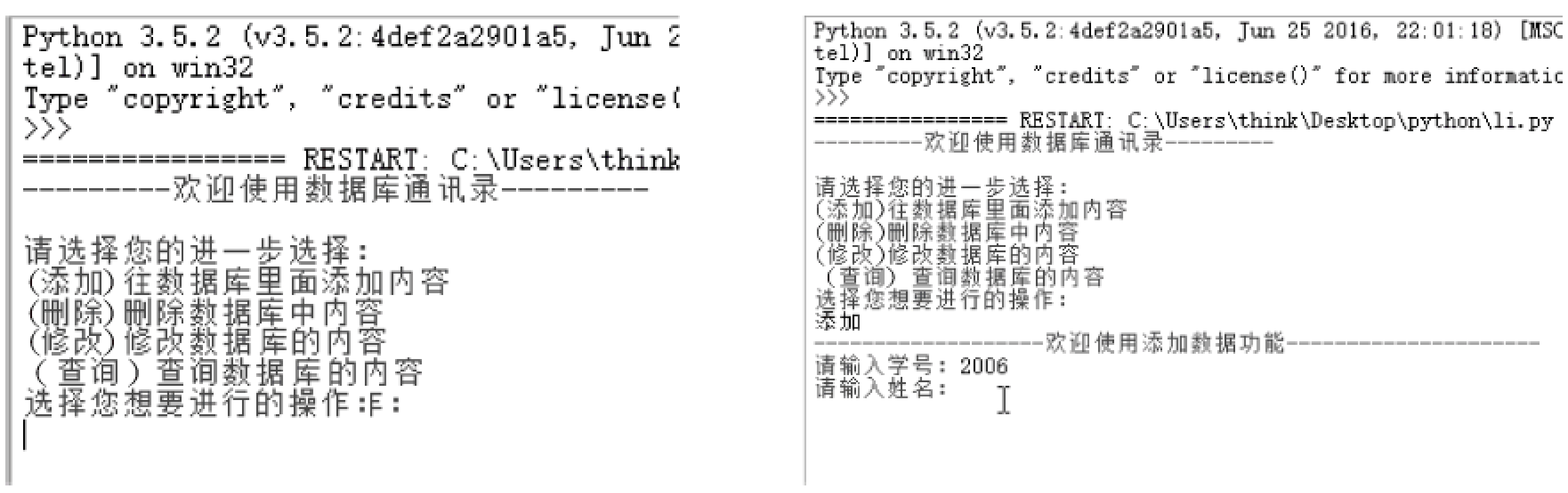


图 3-2　程序运行界面和添加记录界面

3.4　程序设计的步骤

视频讲解

3.4.1　生成试题库

```
import sqlite3                #导入 SQLite 驱动
#连接到 SQLite 数据库，数据库文件是 test.db
#如果文件不存在，会自动在当前目录创建
conn=sqlite3.connect('test2.db')
cursor=conn.cursor()          #创建一个 Cursor
#cursor.execute("delete from exam")
#执行一条 SQL 语句，创建 exam 表
cursor.execute('create table [exam] ([question] varchar(80) null, [Answer A]
varchar(1) null,[Answer B] varchar(1) null,[Answer C] varchar(1) null,
[Answer D] varchar(1) null,[right Answer] varchar(1)  null)')
#继续执行一条 SQL 语句，插入一条记录
cursor.execute("insert into exam (question, Answer A, Answer B, Answer C,
Answer_D, right_Answer) values ('哈雷慧星的平均周期为', '54 年', '56 年', '73
年', '83 年', 'C')")
cursor.execute("insert into exam (question, Answer_A, Answer_B, Answer_C,
```

```
Answer_D, right_Answer) values('夜郎自大中“夜郎”指的是现在哪个地方？', '贵州',
'云南', '广西', '福建', 'A')")
cursor.execute("insert into exam (question, Answer A, Answer B, Answer C,
Answer_D, right_Answer) values('在中国历史上是谁发明了麻药', '孙思邈', '华佗',
'张仲景', '扁鹊', 'B')")
cursor.execute("insert into exam (question, Answer A, Answer B, Answer C,
Answer D, right Answer) values('京剧中花旦是指', '年轻男子', '年轻女子', '年长男子',
'年长女子', 'B')")
cursor.execute("insert into exam (question, Answer A, Answer B, Answer C,
Answer D, right Answer) values('篮球比赛每队几人？', '4', '5', '6', '7', 'B')")
cursor.execute("insert into exam (question, Answer A, Answer B, Answer C,
Answer_D, right_Answer) values('在天愿作比翼鸟，在地愿为连理枝。讲述的是谁的爱情故事？',
'焦钟卿和刘兰芝', '梁山伯与祝英台', '崔莺莺和张生', '杨贵妃和唐明皇', 'D')")
print(cursor.rowcount)          #通过 rowcount 获得插入的行数
cursor.close()                  #关闭 Cursor
conn.commit()                   #提交事务
conn.close()                    #关闭 Connection
```

以上代码完成数据库 test2.db 的建立。下面实现智力问答测试程序功能。

3.4.2　读取试题信息

```
conn=sqlite3.connect('test2.db')
cursor=conn.cursor()
#执行查询语句
cursor.execute('select * from exam')
#获得查询结果集
values=cursor.fetchall()
cursor.close()
conn.close()
```

以上代码完成数据库 test2.db 的试题信息的读取，存储到 values 列表中。

3.4.3　界面和逻辑设计

callNext()用于判断用户选择的正误，正确则加 10 分，错误不加分；并判断用户是否做完，如果没做完，则将下一题的题目信息显示到 timu 标签，4 个选项显示到 radio1～radio4 这 4 个单选按钮上。

```
import tkinter
from tkinter import *
from tkinter.messagebox import *
def callNext():
    global k
    global score
    useranswer=r.get()                  #获取用户的选择
    print(r.get())                      #获取被选中单选按钮变量值
    if useranswer==values[k][5]:
```

```
            showinfo("恭喜","恭喜你对了！")
            score+=10
        else:
            showinfo("遗憾","遗憾你错了！")
        k=k+1
        if k>=len(values):                  #判断用户是否做完
            showinfo("提示","题目做完了")
            return
        #显示下一题
        timu["text"]=values[k][0]           #题目信息
        radio1["text"]=values[k][1]         #A 选项
        radio2["text"]=values[k][2]         #B 选项
        radio3["text"]=values[k][3]         #C 选项
        radio4["text"]=values[k][4]         #D 选项
        r.set('E')

    def callResult():
        showinfo("你的得分",str(score))
```

以下是界面布局代码：

```
    root=tkinter.Tk()
    root.title('Python 智力问答游戏')
    root.geometry("500x200")
    r=tkinter.StringVar()                       #创建 StringVar 对象
    r.set('E')                                  #设置初始值为'E'，初始没选中
    k=0
    score=0
    timu=tkinter.Label(root,text=values[k][0]) #题目
    timu.pack()
    f1=Frame(root)                          #创建第 1 个 Frame 组件
    f1.pack()
    radio1=tkinter.Radiobutton(f1,variable=r,value='A',text=values[k][1])
    radio1.pack()
    radio2=tkinter.Radiobutton(f1,variable=r,value='B',text=values[k][2])
    radio2.pack()
    radio3=tkinter.Radiobutton(f1,variable=r,value='C',text=values[k][3])
    radio3.pack()
    radio4=tkinter.Radiobutton(f1,variable=r,value='D',text=values[k][4])
    radio4.pack()
    f2=Frame(root)                          #创建第 2 个 Frame 组件
    f2.pack()
    Button(f2,text='下一题',command=callNext).pack(side=LEFT)
    Button(f2,text='结  果',command=callResult).pack(side=LEFT)
    root.mainloop()
```

第4章 调用百度API应用——小小翻译器

4.1 小小翻译器功能介绍

视频讲解

小小翻译器使用百度翻译开放平台提供的 API，实现简单的翻译功能，用户输入自己需要翻译的单词或者句子，即可得到翻译的结果，运行界面如图 4-1 所示。该翻译器不仅能够将英文翻译成中文，也可以将中文翻译成英文，或者是其他语言。

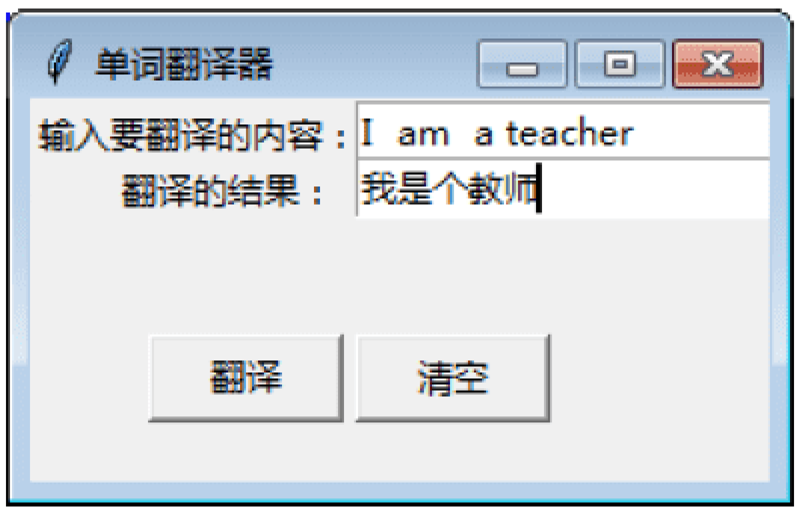


图 4-1　小小翻译器的运行界面

4.2 程序设计的思路

百度翻译开放平台提供的 API，可以为用户提供高质量的翻译服务。通过调用百度翻译 API 编写在线翻译程序。

百度翻译开放平台每月提供 200 万字符的免费翻译服务，只要拥有百度账号并申请成为开发者就可以获得所需要的账号和密码。下面是开发者申请链接：

http://api.fanyi.baidu.com/api/trans/product/index

为方便使用，百度翻译开放平台提供了详细的接入文档，链接如下：

http://api.fanyi.baidu.com/api/trans/product/apidoc

在相应文档中列出了详细的使用方法。

按照百度翻译开放平台文档中的要求，生成 URL 请求网页，提交后可返回 JSON 数据格式的翻译结果，再将得到的 JSON 格式的翻译结果解析出来。

4.3 关键技术

视频讲解

4.3.1 urllib 库简介

urllib 是 Python 标准库中最常用的 Python 网页访问的模块，它可以让用户像访问本地文本文件一样读取网页的内容。Python2 系列使用的是 urllib2，Python3 以后将其全部整合为 urllib；在 Python3.x 中，用户可以使用 urllib 这个库抓取网页。

urllib 库提供了一个网页访问的简单易懂的 API 接口，还包括一些函数方法，用于进行参数编码、下载网页等操作。这个模块的使用门槛非常低，初学者也可以尝试去抓取和读取或者保存网页。urllib 是一个 URL 处理包，在这个包中集合了一些处理 URL 的模块。

（1）urllib.request 模块：用来打开和读取 URL。

（2）urllib.error 模块：包含一些由 urllib.request 产生的错误，可以使用 try 进行捕捉处理。

（3）urllib.parse 模块：包含一些解析 URL 的方法。

（4）urllib.robotparser 模块：用来解析 robots.txt 文本文件。它提供了一个单独的 RobotFileParser 类，通过该类提供的 can_fetch()方法测试爬虫是否可以下载一个页面。

4.3.2 urllib 库的基本使用

下面结合使用 urllib.request 和 urllib.parse 两个模块说明 urllib 库的使用方法。

❶ 获取网页信息

使用 urllib.request.urlopen()函数可以很轻松地打开一个网站，读取并打印网页信息。

urlopen()函数的语法格式如下：

```
urlopen(url[, data[, proxies]])
```

urlopen()返回一个 Response 对象，然后像本地文件一样操作这个 Response 对象来获取远程数据。其中，参数 url 表示远程数据的路径，一般是网址；参数 data 表示以 post 方式提交到 URL 的数据（提交数据有两种方式——post 与 get，一般情况下很少用到这个参数）；参数 proxies 用于设置代理。urlopen()还有一些可选参数，对于具体信息，读者可以查阅 Python 自带的文档。

urlopen()返回的 Response 对象提供了如下方法。

- read()、readline()、readlines()、fileno()、close()：这些方法的使用方式和文件对象完全一样。
- info()：返回一个 httplib.HTTPMessage 对象，表示远程服务器返回的头信息。
- getcode()：返回 HTTP 状态码。如果是 HTTP 请求，200 表示请求成功完成，404 表示网址未找到。
- geturl()：返回请求的 URL。

了解了这些，读者就可以编写一个最简单的爬取网页的程序。

```
#urllib test01.py
from urllib import request
if __name__=="__main__":
    response=request.urlopen("http://fanyi.baidu.com")
    html=response.read()
    html=html.decode("utf-8")  #decode()将网页的信息进行解码，否则会产生乱码
    print(html)
```

urllib 使用 request.urlopen()打开和读取 URL 信息，返回的对象 Response 如同一个文本对象，用户可以调用 read()进行读取，再通过 print()将读到的信息打印出来。

运行 py 程序文件，输出信息如图 4-2 所示。

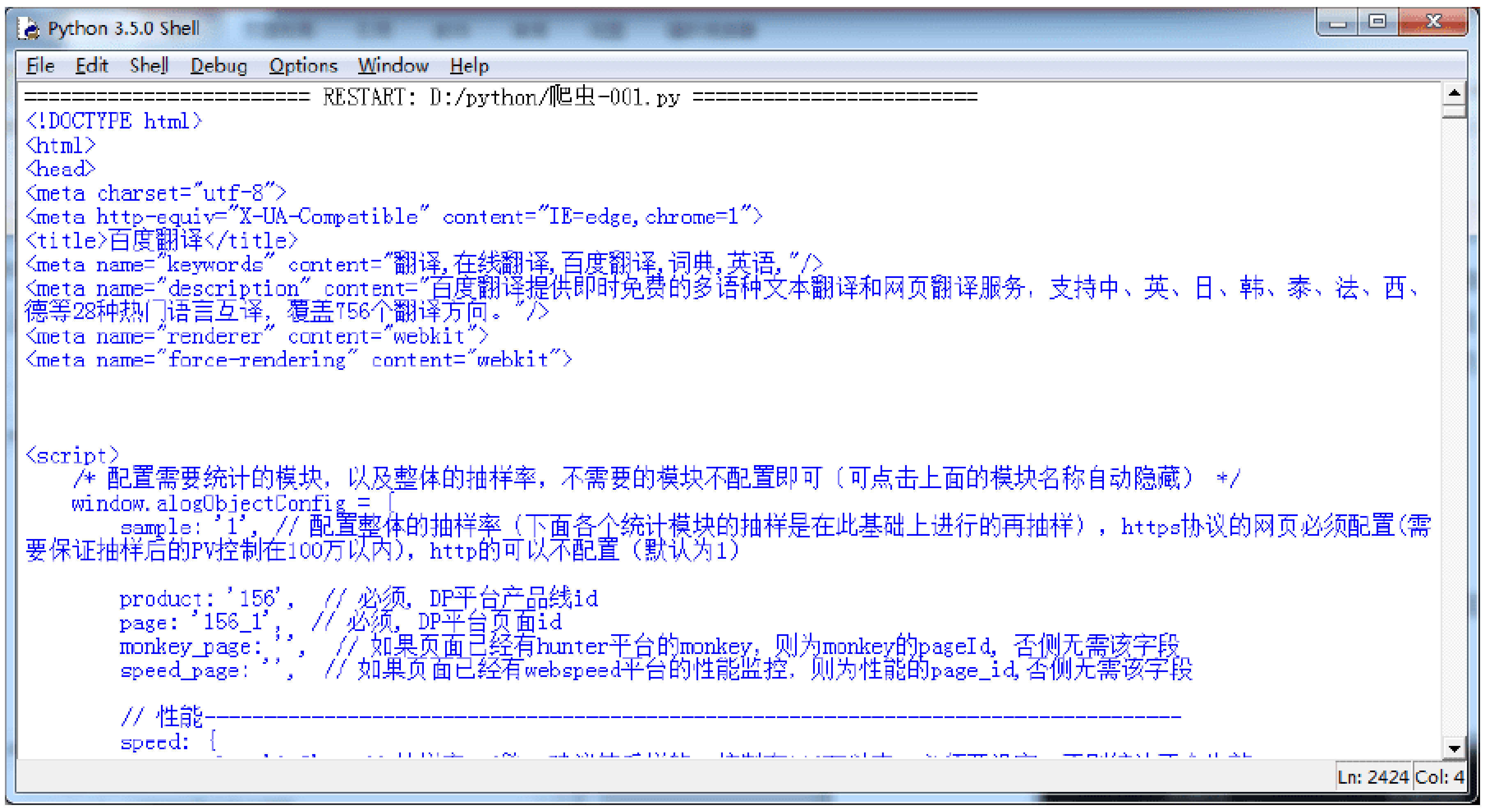


图 4-2　读取的百度翻译网页源码

其实，这就是浏览器接收到的信息，只不过用户在使用浏览器的时候，浏览器已经将这些信息转化成了界面信息供用户浏览。浏览器就是作为客户端从服务器端获取信息，然后将信息解析，再展示给用户的。

这里通过 decode()命令将网页的信息进行解码：

```
html=html.decode("utf-8")
```

当然，这个前提是用户已经知道了这个网页是使用 utf-8 编码的，那么怎么查看网页的

编码方式呢？非常简单的方法是使用浏览器查看网页源码，只需要找到 head 标签开始位置的 chareset，就能知道网页是采用何种编码了。

需要说明的是，urlopen()函数中的 url 参数不仅可以是一个字符串，例如"http://www.baidu.com"，还可以是一个 Request 对象，这就需要先定义一个 Request 对象，然后将这个 Request 对象作为 urlopen()的参数使用，方法如下：

```
req=request.Request("http://fanyi.baidu.com/")   #Request 对象
response=request.urlopen(req)
html=response.read()
html=html.decode("utf-8")
print(html)
```

注意：如果要把对应文件下载到本地，可以使用 urlretrieve()函数。

```
from urllib import request
request.urlretrieve("http://www.zzti.edu.cn/_mediafile/index/2017/06/24
/1qjdyc7vq5.jpg","aaa.jpg")
```

这样就可以把网络上中原工学院的图片资源 1qjdyc7vq5.jpg 下载到本地，生成 aaa.jpg 图片文件。

❷ 获取服务器响应信息

和浏览器的交互过程一样，request.urlopen()代表请求过程，它返回的 HTTPResponse 对象代表响应。返回内容作为一个对象更便于操作，HTTPResponse 对象的 status 属性返回请求 HTTP 后的状态，在处理数据之前要先判断状态情况。如果请求未被响应，需要终止内容处理。reason 属性非常重要，可以得到未被响应的原因，url 属性用于返回页面 URL。HTTPResponse.read()用于获取请求的页面内容的二进制形式。

用户也可以使用 getheaders()返回 HTTP 响应的头信息，例如：

```
from urllib import request
f=request.urlopen('http://fanyi.baidu.com')
data=f.read()
print('Status:', f.status, f.reason)
for k, v in f.getheaders():
   print('%s: %s' % (k, v))
```

可以看到 HTTP 响应的头信息。

```
Status: 200 OK
Content-Type: text/html
Date: Sat, 15 Jul 2017 02:18:26 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Server: Apache
Set-Cookie: locale=zh; expires=Fri, 11-May-2018 02:18:26 GMT; path=/;
domain=.baidu.com
Set-Cookie: BAIDUID=2335F4F896262887F5B2BCEAD460F5E9:FG=1; expires=Sun,
15-Jul-18 02:18:26 GMT; max-age=31536000; path=/; domain=.baidu.com;
```

```
version=1
Vary: Accept-Encoding
Connection: close
Transfer-Encoding: chunked
```

同样可以使用 Response 对象的 geturl()方法、info()方法、getcode()方法获取相关的 URL、响应信息和响应 HTTP 状态码。

```
#-*- coding: UTF-8 -*-
from urllib import request
if __name__=="__main__":
    req=request.Request("http://fanyi.baidu.com/")
    response=request.urlopen(req)
    print("geturl 打印信息: %s"%(response.geturl()))
    print('**********************************************')
    print("info 打印信息: %s"%(response.info()))
    print('**********************************************')
    print("getcode 打印信息: %s"%(response.getcode()))
```

可以得到如下运行结果：

```
geturl 打印信息: http://fanyi.baidu.com/
**********************************************
info 打印信息: Content-Type: text/html
Date: Sat, 15 Jul 2017 02:42:32 GMT
P3p: CP=" OTI DSP COR IVA OUR IND COM "
Server: Apache
Set-Cookie: locale=zh; expires=Fri, 11-May-2018 02:42:32 GMT; path=/;
domain=.baidu.com
Set-Cookie: BAIDUID=976A41D6B0C3FD6CA816A09BEAC3A89A:FG=1; expires=Sun,
15-Jul-18 02:42:32 GMT; max-age=31536000; path=/; domain=.baidu.com;
version=1
Vary: Accept-Encoding
Connection: close
Transfer-Encoding: chunked
**********************************************
getcode 打印信息: 200
```

现在读者已经学会了使用简单的语句对网页进行抓取，接下来学习如何向服务器发送数据。

❸ 向服务器发送数据

用户可以使用 urlopen()函数中的 data 参数向服务器发送数据。根据 HTTP 规范，get 用于信息获取，post 是向服务器提交数据的一种请求。换句话说，从客户端向服务器提交数据使用 post，从服务器获得数据到客户端使用 get。get 也可以提交，与 post 的区别如下：

（1）get 方式可以通过 URL 提交数据，待提交数据是 URL 的一部分；采用 post 方式，待提交数据放置在 HTML header 内。

（2）get 方式提交的数据最多不超过 1024 个字节，post 没有对提交内容的长度做限制。

如果没有设置 urlopen()函数的 data 参数，HTTP 请求采用 get 方式，也就是从服务器获取信息；如果设置 data 参数，HTTP 请求采用 post 方式，也就是向服务器传递数据。

data 参数有自己的格式，它是一个基于 application/x-www.form-urlencoded 的格式，对于其具体格式，读者不用了解，因为可以使用 urllib.parse.urlencode()函数将字符串自动转换成上面所说的格式。

❹ 使用 User Agent 隐藏身份

1）为何要设置 User Agent

有些网站不喜欢被爬虫程序访问，所以会检测连接对象，如果是爬虫程序，也就是非人点击访问，它就会不让继续访问，所以为了让程序可以正常运行，需要隐藏自己的爬虫程序的身份。此时可以通过设置 User Agent 来达到隐藏身份的目的，User Agent 的中文名为用户代理，简称 UA。

User Agent 存放于 headers 中，服务器就是通过查看 headers 中的 User Agent 来判断是谁在访问。在 Python 中，如果不设置 User Agent，程序将使用默认的参数，那么这个 User Agent 就会有 Python 的字样，如果服务器检查 User Agent，那么没有设置 User Agent 的 Python 程序将无法正常访问网站。

Python 允许用户修改这个 User Agent 来模拟浏览器访问，它的强大毋庸置疑。

2）常见的 User Agent

（1）Android：

```
•Mozilla/5.0 (Linux; Android 4.1.1; Nexus 7 Build/JRO03D) AppleWebKit/535.19
(KHTML, like Gecko) Chrome/18.0.1025.166 Safari/535.19
•Mozilla/5.0 (Linux; U; Android 4.0.4; en-gb; GT-I9300 Build/IMM76D)
AppleWebKit/534.30 (KHTML, like Gecko) Version/4.0 Mobile Safari/534.30
•Mozilla/5.0 (Linux; U; Android 2.2; en-gb; GT-P1000 Build/FROYO)
AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1
```

（2）Firefox：

```
•Mozilla/5.0 (Windows NT 6.2; WOW64; rv:21.0) Gecko/20100101 Firefox/21.0
•Mozilla/5.0 (Android; Mobile; rv:14.0) Gecko/14.0 Firefox/14.0
```

（3）Google Chrome：

```
•Mozilla/5.0 (Windows NT 6.2; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/27.0.1453.94 Safari/537.36
•Mozilla/5.0 (Linux; Android 4.0.4; Galaxy Nexus Build/IMM76B) AppleWebKit/
535.19 (KHTML, like Gecko) Chrome/18.0.1025.133 Mobile Safari/535.19
```

（4）iOS：

```
•Mozilla/5.0 (iPad; CPU OS 5_0 like Mac OS X) AppleWebKit/534.46 (KHTML, like
Gecko) Version/5.1 Mobile/9A334 Safari/7534.48.3
•Mozilla/5.0 (iPod; U; CPU like Mac OS X; en) AppleWebKit/420.1 (KHTML, like
Gecko) Version/3.0 Mobile/3A101a Safari/419.3
```

上面列举了 Android、Firefox、Google Chrome、iOS 的一些 User Agent。

3）设置 User Agent 的方法

设置 User Agent 有以下两种方法：

（1）在创建 Request 对象的时候填入 headers 参数（包含 User Agent 信息），这个 headers 参数要求为字典。

（2）在创建 Request 对象的时候不添加 headers 参数，在创建完成之后使用 add_header()方法添加 headers。

方法一：

使用上面提到的 Android 的第 1 个 User Agent，在创建 Request 对象的时候传入 headers 参数，编写代码如下：

```
#-*- coding: UTF-8 -*-
from urllib import request
if __name__=="__main__":
    #以 CSDN 为例，CSDN 不更改 User Agent 是无法访问的
    url='http://www.csdn.net/'
    head={}
    #写入 User Agent 信息
    head['User-Agent']='Mozilla/5.0 (Linux; Android 4.1.1; Nexus 7 Build/
    JRO03D) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.166
    Safari/535.19'
    req=request.Request(url, headers=head)       #创建 Request 对象
    response=request.urlopen(req)                #传入创建好的 Request 对象
    html=response.read().decode('utf-8')         #读取响应信息并解码
    print(html)                                  #打印信息
```

方法二：

使用上面提到的 Android 的第 1 个 User Agent，在创建 Request 对象时不传入 headers 参数，在创建之后使用 add_header()方法添加 headers，编写代码如下：

```
#-*- coding: UTF-8 -*-
from urllib import request
if __name__=="__main__":
    #以 CSDN 为例，CSDN 不更改 User Agent 是无法访问的
    url='http://www.csdn.net/'
    req=request.Request(url)                     #创建 Request 对象
    req.add header('User-Agent', 'Mozilla/5.0 (Linux; Android 4.1.1; Nexus
    7 Build/JRO03D) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/
    18.0.1025.166 Safari/535.19')                #传入 headers
    response=request.urlopen(req)                #传入创建好的 Request 对象
    html=response.read().decode('utf-8')         #读取响应信息并解码
    print(html)                                  #打印信息
```

4.4 程序设计的步骤

视频讲解

4.4.1 设计界面

采用 Tkinter 的 place 几何布局管理器设计 GUI 图形界面，运行效果如图 4-3 所示。

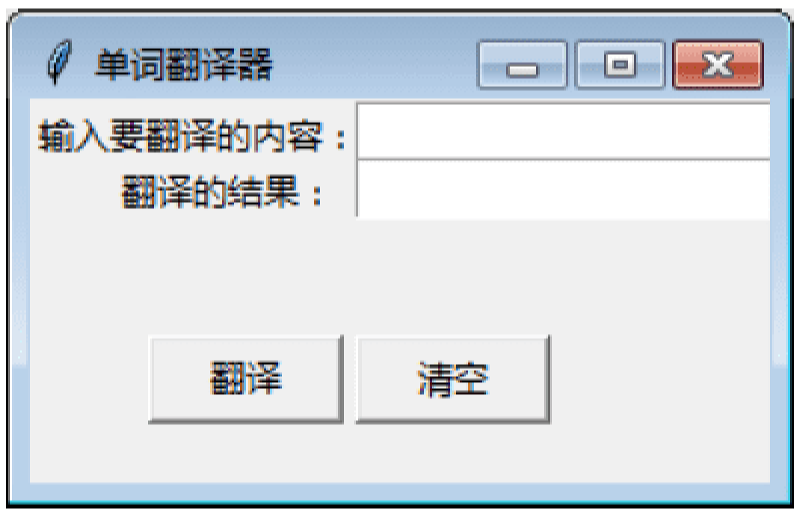


图 4-3　place 几何布局管理器

新建文件 translate_test.py，编写如下代码：

```
from tkinter import *
if __name__=="__main__":
    root=Tk()
    root.title("单词翻译器")
    root['width']=250;root['height']=130
    Label(root,text='输入要翻译的内容：',width=15).place(x=1,y=1)
                                            #绝对坐标(1, 1)
    Entry1=Entry(root,width=20)
    Entry1.place(x=110,y=1)                 #绝对坐标(110, 1)
    Label(root,text='翻译的结果：',width=18).place(x=1,y=20) #绝对坐标(1, 20)
    s=StringVar()                           #一个 StringVar()对象
    s.set("大家好，这是测试")
    Entry2=Entry(root,width=20,textvariable=s)
    Entry2.place(x=110,y=20)                #绝对坐标(110, 20)
    Button1=Button(root,text='翻译',width=8)
    Button1.place(x=40,y=80)                #绝对坐标(40, 80)
    Button2=Button(root,text='清空',width=8)
    Button2.place(x=110,y=80)               #绝对坐标(110, 80)
    #给 Button 绑定鼠标监听事件
    Button1.bind("<Button-1>",leftClick)    #“翻译”按钮
    Button2.bind("<Button-1>",leftClick2)   #“清空”按钮
    root.mainloop()
```

4.4.2 使用百度翻译开放平台 API

使用百度翻译需要向“http://api.fanyi.baidu.com/api/trans/vip/translate”地址通过 post 或 get 方法发送表 4-1 中的请求参数来访问服务。

表 4-1　请求参数

参数名	类型	必填参数	描述	备注
q	TEXT	Y	请求翻译 query	UTF-8 编码
from	TEXT	Y	翻译源语言	语言列表（可设置为 auto）
to	TEXT	Y	译文语言	语言列表（不可设置为 auto）
appid	INT	Y	APP ID	可在管理控制台查看
salt	INT	Y	随机数	
sign	TEXT	Y	签名	appid+q+salt+密钥的 md5 值

sign 签名是为了保证调用安全，使用 md5 算法生成的一段字符串，生成的签名长度为 32 位，签名中的英文字符均为小写格式。为保证翻译质量，请将单次请求长度控制在 6000 字节以内（汉字约为 2000 个）。

签名的生成方法如下：

（1）将请求参数中的 appid、query（q，注意为 UTF-8 编码）、随机数 salt 以及平台分配的密钥（可在管理控制台查看）按照 appid+q+salt+密钥的顺序拼接得到字符串 1。

（2）对字符串 1 做 md5，得到 32 位小写的 sign。

注意：

（1）先将需要翻译的文本转换为 UTF-8 编码。

（2）在发送 HTTP 请求之前需要对各字段做 URL encode。

（3）在生成签名拼接字符串时，q 不需要做 URL encode，在生成签名之后发送 HTTP 请求之前才需要对要发送的待翻译文本字段 q 做 URL encode。

例如将 apple 从英文翻译成中文。

请求参数：

```
q=apple
from=en
to=zh
appid=2015063000000001
salt=1435660288
平台分配的密钥：12345678
```

生成签名参数 sign：

（1）拼接字符串 1。

```
拼接 appid=2015063000000001+q=apple+salt=1435660288+密钥=12345678
得到字符串 1=2015063000000001apple143566028812345678
```

（2）计算签名 sign（对字符串 1 做 md5 加密，注意在计算 md5 之前，字符串 1 必须为 UTF-8 编码）。

```
sign=md5(2015063000000001apple143566028812345678)
sign=f89f9594663708c1605f3d736d01d2d4
```

通过 Python 提供的 hashlib 模块中的 hashlib.md5()可以实现签名计算。例如：

```
import hashlib
m='2015063000000001apple143566028812345678'
m_MD5=hashlib.md5(m)
sign=m_MD5.hexdigest()
print( 'm=',m)
print('sign=',sign)
```

在得到签名之后，按照百度文档中的要求生成 URL 请求，提交后可返回翻译结果。

其完整请求如下：

http://api.fanyi.baidu.com/api/trans/vip/translate?q=apple&from=en&to=zh&appid=2015063000000001&salt=1435660288&sign=f89f9594663708c1605f3d736d01d2d4

用户也可以使用 post 方法传送需要的参数。

本例采用 urllib.request.urlopen()函数中的 data 参数向服务器发送数据。

下面是发送 data 的实例，向“百度翻译”发送要翻译数据 data，得到翻译结果。

```
#-*- coding: UTF-8 -*-
from tkinter import *
from urllib import request
from urllib import parse
import json
import hashlib
def translate_Word(en_str):
    #simulation browse load host url,get cookie
    URL='http://api.fanyi.baidu.com/api/trans/vip/translate'
    #en_str=input("请输入要翻译的内容:")
    #创建 Form Data 字典，存储向服务器发送的 data
    #Form_Data={'from':'en','to':'zh','q':en_str,"appid":
    '2015063000000001','salt':'1435660288'}
    Form_Data={}
    Form_Data['from']='en'
    Form_Data['to']='zh'
    Form_Data['q']=en_str                         #要翻译数据
    Form_Data['appid']='2015063000000001'   #申请的 APP ID
    Form_Data['salt']='1435660288'
    Key="12345678"                                #平台分配的密钥
    m=Form_Data['appid']+en_str+Form_Data['salt']+Key
    m_MD5=hashlib.md5(m.encode('utf8'))
    Form_Data['sign']=m_MD5.hexdigest()

    data=parse.urlencode(Form_Data).encode('utf-8')
                                                  #使用 urlencode()方法转换标准格式
    response=request.urlopen(URL,data)         #传递 Request 对象和转换完格式的数据
    html=response.read().decode('utf-8')      #读取信息并解码
    translate_results=json.loads(html)         #使用 JSON
```

```
    print(translate_results)                     #打印出 JSON 数据
    translate_results=translate_results['trans_result'][0]['dst']
                                                 #找到翻译结果
    print("翻译的结果是：%s" % translate_results) #打印翻译信息
    return translate_results
def leftClick(event):                            #“翻译”按钮事件函数
   en_str=Entry1.get()                           #获取要翻译的内容
   print(en_str)
   vText=translate_Word(en_str)
   Entry2.config(Entry2,text=vText)              #修改翻译结果框文字
   s.set("")
   Entry2.insert(0,vText)
def leftClick2(event):                           #“清空”按钮事件函数
   s.set("")
   Entry2.insert(0,"")
```

这样就可以查看翻译的结果了，如下：

```
输入要翻译的内容：I  am  a  teacher
翻译的结果是：我是个教师。
```

此时得到的 JSON 数据如下：

```
{'from': 'en', 'to': 'zh', 'trans_result': [{'dst': '我是个教师。', 'src':
'I  am  a  teacher'}]}
```

其返回结果是 JSON 格式，包含表 4-2 中的字段。

表 4-2　翻译结果的 JSON 字段

字 段 名	类　型	描　述
from	TEXT	翻译源语言
to	TEXT	译文语言
trans_result	MIXED LIST	翻译结果
src	TEXT	原文
dst	TEXT	译文

trans_result 中包含了 src 和 dst 字段。

JSON 是一种轻量级的数据交换格式，其中保存了用户想要的翻译结果，需要从爬取到的内容中找到 JSON 格式的数据，再将得到的 JSON 格式的翻译结果解析出来。

这里向服务器发送数据 Form_Data 也可以直接如下写：

```
Form_Data={'from':'en',  'to':'zh',  'q':en_str,"appid":
'2015063000000001', 'salt': '1435660288'}
```

现在只是将英文翻译成中文，稍微改一下就可以将中文翻译成英文了：

```
Form_Data={'from':'zh',  'to':'en', 'q':en_str,''appid'':
'2015063000000001', 'salt': '1435660288'}
```

这一行中的 from 和 to 的取值应该可以用于其他语言之间的翻译。如果源语言语种不

确定可设置为 auto，注意目标语言语种不可设置为 auto。百度翻译支持的语言简写如表 4-3 所示。

表 4-3　百度翻译支持的语言简写

语 言 简 写	名　　称	语 言 简 写	名　　称
auto	自动检测	bul	保加利亚语
zh	中文	est	爱沙尼亚语
en	英语	dan	丹麦语
yue	粤语	fin	芬兰语
wyw	文言文	cs	捷克语
jp	日语	rom	罗马尼亚语
kor	韩语	slo	斯洛文尼亚语
fra	法语	swe	瑞典语
spa	西班牙语	hu	匈牙利语
th	泰语	cht	繁体中文
ara	阿拉伯语	vie	越南语
ru	俄语	el	希腊语
pt	葡萄牙语	nl	荷兰语
de	德语	pl	波兰语

请读者查阅资料编程，向有道翻译（http://fanyi.youdao.com/translate?smartresult=dict）发送要翻译数据 data，得到翻译结果。

第 5 章

爬虫应用——校园网搜索引擎

5.1 校园网搜索引擎功能分析

视频讲解

随着校园网建设的迅速发展，校园网内的信息内容正在以惊人的速度增加着。如何更全面、更准确地获取最新、最有效的信息已经成为人们把握机遇、迎接挑战和获取成功的重要条件。目前虽然已经有了像 Google、百度这样优秀的通用搜索引擎，但是它们并不能适用于所有的情况和需要。对学术搜索、校园网的搜索来说，一个合理的排序结果是非常重要的。另外，互联网上的信息量巨大，远远超出哪怕是最大的一个搜索引擎可以完全收集的能力范围。本章旨在使用 Python 建立一个适合校园网使用的 Web 搜索引擎系统，它能在较短的时间内爬取页面信息，具有有效、准确的中文分词功能，能实现对校园网上新闻信息的快速检索展示。

5.2 校园网搜索引擎系统设计

校园网搜索引擎一般需要以下几个步骤：

（1）网络爬虫爬取这个网站，得到所有网页链接。

网络爬虫就是一只会嗅着 URL（链接）爬过成千上万个网页，并把网页内容搬到用户计算机上供用户使用的苦力虫子。如图 5-1 所示，给定爬虫的出发页面 A 的 URL，它就从起始页 A 出发，读取 A 的所有内容，并从中找到 5 个 URL，分别指向页面 B、C、D、E 和 F，然后它顺着链接依次抓取 B、C、D、E 和 F 页面的内容，并从中发现新的链接，再沿着链接爬到新的页面，对爬虫带回来的网页内容分析链接，继续爬到新的页面，以此类推，直到找不到新的链接或者满足了人为设定的停止条件为止。

至于这只虫子前进的方式，则分为广度优先搜索（BFS）和深度优先搜索（DFS）。在这张图中 BFS 的搜索顺序是 A-B-C-D-E-F-G-H-I，而深度优先搜索的顺序是遍历的路径，

即 A-F-G　E-H-I　B C D。

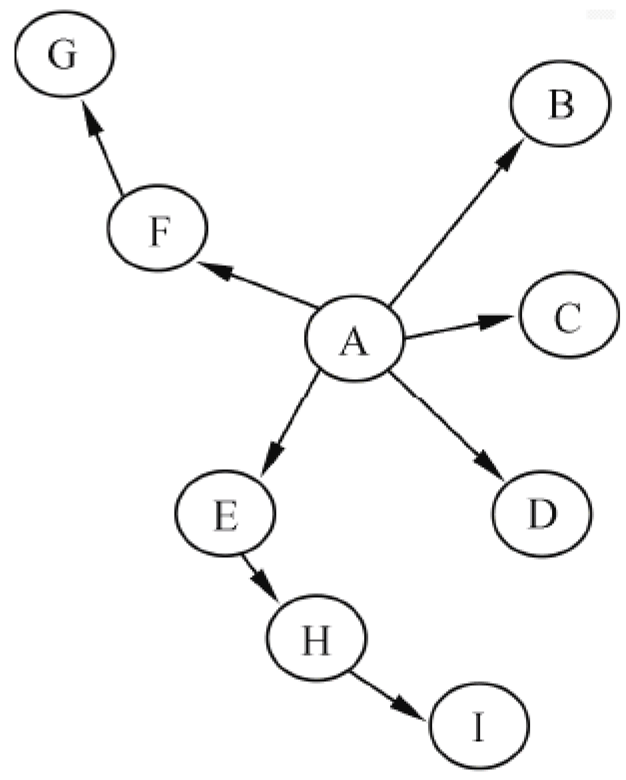


图 5-1　网站链接示意图

（2）得到网页的源代码，解析剥离出想要的新闻内容、标题、作者等信息。

（3）把所有网页的新闻内容做成词条索引，一般采用倒排表索引。

索引一般有正排索引（正向索引）和倒排索引（反向索引）两种类型。

- 正排索引（正向索引，forward index）：正排表是以文档的 ID 为关键字，表中记录文档（即网页）中每个字或词的位置信息，查找时扫描表中每个文档中字或词的信息直到找出所有包含查询关键字的文档。

正排表的结构如图 5-2 所示，这种组织方法在建立索引的时候结构比较简单，建立比较方便且易于维护；因为索引是基于文档建立的，若是有新的文档加入，直接为该文档建立一个新的索引块，挂接在原来索引文件的后面。若是有文档删除，则直接找到该文档号文档对应的索引信息，将其直接删除。但是在查询的时候需要对所有的文档进行扫描以确保没有遗漏，这样就使得检索时间大大延长，检索效率低下。

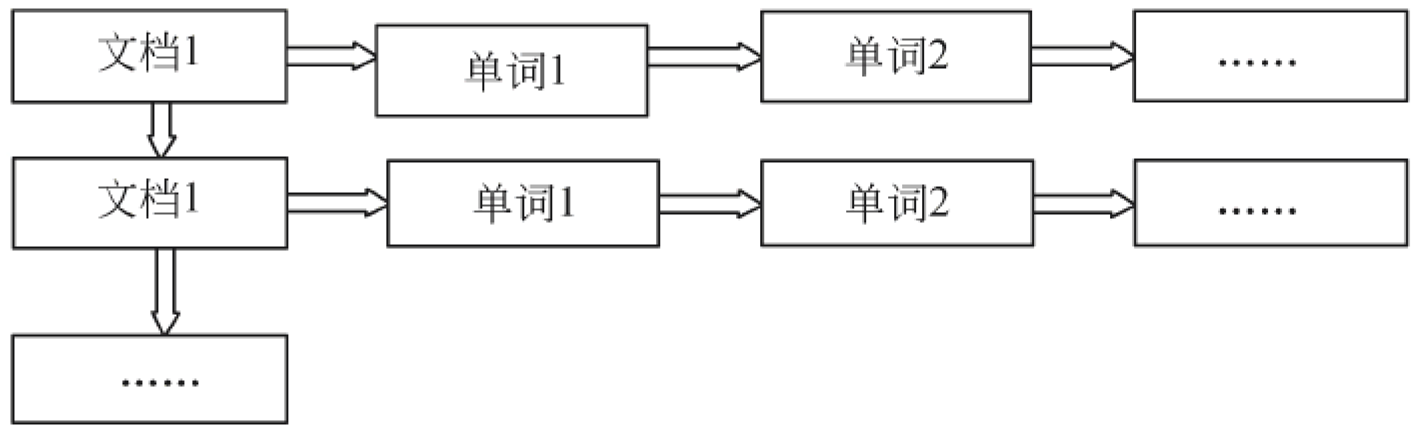


图 5-2　正排表结构示意图

尽管正排表的工作原理非常简单，但是由于其检索效率太低，除非在特定情况下，否则实用价值不大。

- 倒排索引（反向索引，inverted index）：倒排表以字或词为关键字进行索引，表中关键字所对应的记录表项记录了出现这个字或词的所有文档，一个表项就是一个字表段，它记录该文档的 ID 和字符在该文档中出现的位置情况。

由于每个字或词对应的文档数量在动态变化，所以倒排表的建立和维护都较为复杂，但是在查询的时候由于可以一次得到查询关键字所对应的所有文档，所以效率高于正排表。

在全文检索中，检索的快速响应是一个最为关键的性能，而索引的建立由于在后台进行，尽管效率相对低一些，但不会影响整个搜索引擎的效率。

倒排表的结构如图 5-3 所示。

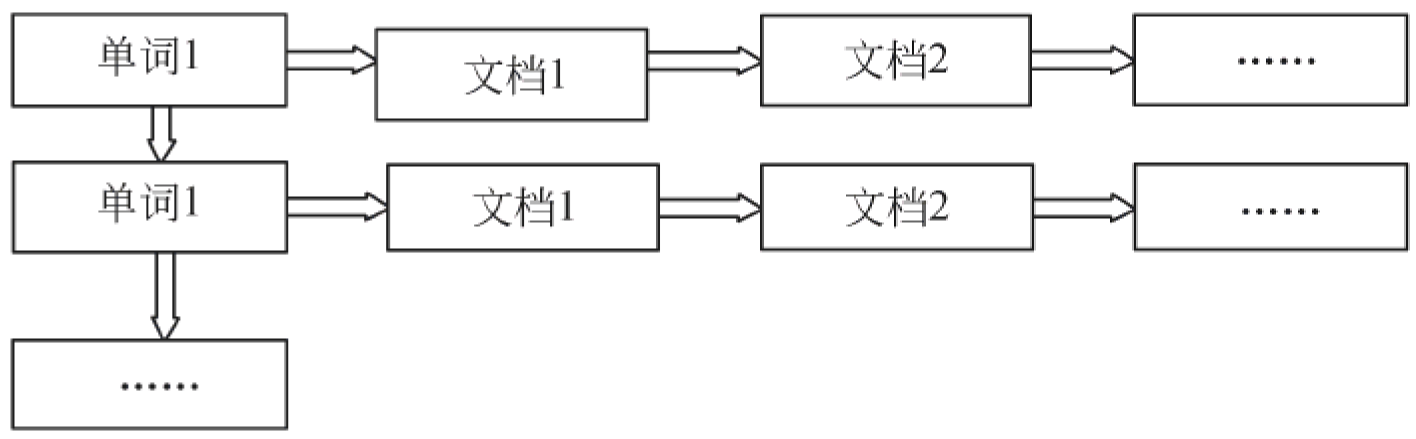


图 5-3　倒排表结构示意图

正排索引是从文档到关键字的映射（已知文档求关键字），倒排索引是从关键字到文档的映射（已知关键字求文档）。

在搜索引擎中每个文件都对应一个文件 ID，文件内容被表示为一系列关键词的集合（实际上，在搜索引擎索引库中关键词已经转换为关键词 ID）。例如“文档 1”经过分词提取了 20 个关键词，每个关键词都会记录它在文档中的出现次数和出现位置，得到正向索引的结构如下：

```
“文档 1”的 ID > 单词 1：出现次数，出现位置列表；单词 2：出现次数，出现位置列表；……
“文档 2”的 ID > 此文档出现的关键词列表
```

当用户搜索关键词“华为手机”时，假设只存在正向索引（forward index），那么就需要扫描索引库中的所有文档，找出所有包含关键词“华为手机”的文档，再根据打分模型进行打分，排出名次后呈现给用户。因为互联网上收录在搜索引擎中的文档的数目是个天文数字，这样的索引结构根本无法满足实时返回排名结果的要求。所以搜索引擎会将正向索引重新构建为倒排索引，即把文件 ID 对应到关键词的映射转换为关键词到文件 ID 的映射，每个关键词都对应一系列的文件，这些文件中都出现这个关键词，得到倒排索引的结构如下：

```
“单词 1”：“文档 1”的 ID，“文档 2”的 ID，……
“单词 2”：带有此关键词的文档 ID 列表
```

（4）搜索时，根据搜索词在词条索引中查询，按顺序返回相关的搜索结果，也可以按网页评价排名顺序返回相关的搜索结果。

当用户输入一串搜索字符串时，程序会先进行分词，然后依照每个词的索引找到相应网页。假如在搜索框中输入“从前有座山山里有座庙小和尚”，搜索引擎首先会对字符串进行分词处理“从前/有/座山/山里/有/座庙/小和尚”，然后按照一定的规则对词做布尔运算，例如每个词之间做“与”运算，在索引中搜索“同时”包含这些词的页面。

所以本系统主要由以下 4 个模块组成。

- 信息采集模块：主要是利用网络爬虫实现对校园网信息的抓取。
- 索引模块：负责对爬取的新闻网页的标题、内容和作者进行分词并建立倒排词表。

- 网页排名模块：TF/IDF 是一种统计方法，用于评估一字词对于一个文件集或一个语料库中的一份文件的重要程度。
- 用户搜索界面模块：负责用户关键字的输入以及搜索结果信息的返回。

5.3 关键技术

5.3.1 正则表达式

把网页中的超链接提取出来需要使用正则表达式。那么什么是正则表达式？在回答这个问题之前先来看一看为什么要有正则表达式。

在编程处理文本的过程中，经常需要按照某种规则去查找一些特定的字符串。例如知道一个网页上的图片都叫“image/8554278135.jpg”之类的名字，只是那串数字不一样；又或者在一堆人员的电子档案中，要把他们的电话号码全部找出来，整理成通讯录。诸如此类工作，可不可以利用这些规律，让程序自动来做这些事情？答案是肯定的。这时候就需要一种描述这些规律的方法，正则表达式就是描述文本规则的代码。

正则表达式是一种用来匹配字符串文本的强有力的武器。它是用一种描述性的语言来给字符串定义一个规则。凡是符合规则的字符串，就认为它“匹配”了，否则该字符串就是不合法的。

❶ 正则表达式的语法

正则表达式并不是 Python 中特有的功能，它是一种通用的方法，要使用它必须会用正则表达式来描述文本规则。

正则表达式使用特殊的语法来表示，表 5-1 列出了正则表达式的语法。

表 5-1 正则表达式的语法

模　式	描　述
^	匹配字符串的开头
$	匹配字符串的末尾
.	匹配任意字符，除了换行符
[...]	用来表示一组字符，例如[amk]匹配'a'、'm'或'k'；[0-9]匹配任何数字，类似于[0123456789]；[a-z]匹配任何小写字母；[a-zA-Z0-9]匹配任何字母及数字
[^...]	不在[]中的字符，例如[^abc]匹配除了 a、b、c 之外的字符；[^0-9]匹配除了数字之外的字符
*	数量词，匹配 0 个或多个
+	数量词，匹配 1 个或多个
?	数量词，以非贪婪方式匹配 0 个或 1 个
{ n,}	重复 n 次或更多次
{ n, m}	重复 n～m 次
a|b	匹配 a 或 b
(re)	匹配括号内的表达式，也表示一个组
(?imx)	正则表达式包含 3 种可选标志，即 i、m、x，只影响括号中的区域

续表

模　式	描　述
(?-imx)	正则表达式关闭 i、m 或 x 可选标志，只影响括号中的区域
(?: re)	类似(...)，但是不表示一个组
(?imx: re)	在括号中使用 i、m 或 x 可选标志
(?-imx: re)	在括号中不使用 i、m 或 x 可选标志
(?= re)	前向肯定界定符，如果所含正则表达式以…表示，在当前位置成功匹配时成功，否则失败。一旦所含表达式已经尝试，匹配引擎根本没有提高，模式的剩余部分还要尝试界定符的右边
(?! re)	前向否定界定符，与前面肯定界定符相反，当所含表达式不能在字符串当前位置匹配时成功
(?> re)	匹配的独立模式，省去回溯
\w	匹配字母、数字及下画线，等价于'[A-Za-z0-9_]'
\W	匹配非字母、数字及下画线，等价于 '[^A-Za-z0-9_]'
\s	匹配任何空白字符，包括空格、制表符、换页符等，等价于[\f\n\r\t\v]
\S	匹配任何非空白字符，等价于[^ \f\n\r\t\v]
\d	匹配任意数字，等价于[0-9]
\D	匹配任意非数字，等价于[^0-9]
\A	匹配字符串开始
\Z	匹配字符串结束，如果存在换行，只匹配到换行前的结束字符串
\z	匹配字符串结束
\G	匹配最后匹配完成的位置
\b	匹配一个单词边界，也就是单词和空格间的位置。例如，'er\b'可以匹配"never"中的'er'，但不能匹配"verb"中的'er'
\B	匹配非单词边界，例如'er\B'能匹配"verb"中的'er'，但不能匹配"never"中的'er'
\n、\t 等	匹配一个换行符、一个制表符等

正则表达式通常用于在文本中查找匹配的字符串。在 Python 中数量词默认是贪婪的，总是尝试匹配尽可能多的字符；非贪婪的则相反，总是尝试匹配尽可能少的字符。例如，正则表达式"ab*"如果用于查找"abbbc"，将找到"abbb"；如果使用非贪婪的数量词"ab*?"，将找到"a"。

在正则表达式中，如果直接给出字符，就是精确匹配。从正则表达式语法中能够了解到用\d 可以匹配一个数字，用\w 可以匹配一个字母或数字，用.可以匹配任意字符，所以模式'00\d'可以匹配'007'，但无法匹配'00A'；模式'\d\d\d'可以匹配'010'；模式'\w\w\d'可以匹配'py3'；模式'py.'可以匹配'pyc'、'pyo'、'py!'，等等。

如果要匹配变长的字符，在正则表达式模式字符串中用*表示任意个字符（包括 0 个），用+表示至少一个字符，用?表示 0 个或 1 个字符，用{n}表示 n 个字符，用{n,m}表示 n～m 个字符。这里看一个复杂的表示电话号码的例子，即\d{3}\s+\d{3,8}。

从左到右解读一下：

\d{3}表示匹配 3 个数字，例如'010'；

\s 可以匹配一个空格（也包括 Tab 等空白符），所以\s+表示至少有一个空格；

\d{3,8}表示 3～8 个数字，例如'67665230'。

综合起来，上面的正则表达式可以匹配以任意个空格隔开的带区号的电话号码。

如果要匹配'010-67665230'这样的号码，怎么办？由于'-'是特殊字符，在正则表达式中要用'\'转义，所以上面的正则表达式是\d{3}\-\d{3,8}。

如果要做更精确的匹配，可以用[]表示范围，例如：

[0-9a-zA-Z_]可以匹配一个数字、字母或者下画线；

[0-9a-zA-Z_]+可以匹配至少由一个数字、字母或者下画线组成的字符串，例如'a100'、'0_Z'、'Py3000'等；

[a-zA-Z_][0-9a-zA-Z_]*可以匹配以字母或下画线开头，后接任意个由一个数字、字母或者下画线组成的字符串，也就是 Python 合法的变量；

[a-zA-Z_][0-9a-zA-Z_]{0, 19}更精确地限制了变量的长度是 1～20 个字符（前面 1 个字符+后面最多 19 个字符）。

另外，A|B 可以匹配 A 或 B，所以(P|p)ython 可以匹配'Python'或者'python'。

^表示行的开头，^\d 表示必须以数字开头。

$表示行的结束，\d$表示必须以数字结束。

❷ re 模块

Python 提供了 re 模块，包含所有正则表达式的功能。

1）match()方法

re.match()的格式为 re.match(pattern, string, flags)。

其第 1 个参数是正则表达式，第 2 个参数表示要匹配的字符串；第 3 个参数是标志位，用于控制正则表达式的匹配方式，例如是否区分大小写、多行匹配等。

match()方法判断是否匹配，如果匹配成功，返回一个 Match 对象，否则返回 None。常见的判断方法如下：

```
test='用户输入的字符串'
if re.match(r'正则表达式', test):    #r 前缀为原义字符串，它表示对字符串不进行转义
    print('ok')
else:
    print('failed')
```

例如：

```
>>> import re
>>> re.match(r'^\d{3}\-\d{3,8}$', '010-12345')  #返回一个 Match 对象
< sre.SRE Match object; span=(0,9), match='010-12345'>
>>> re.match(r'^\d{3}\-\d{3,8}$', '010 12345')
                                        #'010 12345'不匹配规则，返回 None
```

Match 对象是一次匹配的结果，包含了很多关于此次匹配的信息，可以使用 Match 提供的可读属性或方法来获取这些信息。

Match 属性如下：

- string：匹配时使用的文本。
- re：匹配时使用的 Pattern 对象。
- pos：文本中正则表达式开始搜索的索引，值与 Pattern.match()和 Pattern.search()方法

的同名参数相同。

- endpos：文本中正则表达式结束搜索的索引，值与 Pattern.match()和 Pattern.search()方法的同名参数相同。
- lastindex：最后一个被捕获的分组在文本中的索引。如果没有被捕获的分组，将为 None。
- lastgroup：最后一个被捕获的分组的别名。如果这个分组没有别名或者没有被捕获的分组，将为 None。

Match 方法如下：

- group([group1, …])：获得一个或多个分组截获的字符串，当指定多个参数时将以元组形式返回。参数 group1 可以使用编号也可以使用别名；编号 0 代表整个匹配的子串；当不填写参数时返回 group(0)；没有截获字符串的组返回 None；截获了多次的组返回最后一次截获的子串。
- groups([default])：以元组形式返回全部分组截获的字符串，相当于调用 group(1,2,…, last)。default 表示没有截获字符串的组以这个值替代，默认为 None。
- groupdict([default])：返回以有别名的组的别名为键、以该组截获的子串为值的字典，没有别名的组不包含在内。default 的含义同上。
- start([group])：返回指定的组截获的子串在 string 中的起始索引（子串第 1 个字符的索引）。group 的默认值为 0。
- end([group])：返回指定的组截获的子串在 string 中的结束索引（子串最后一个字符的索引+1）。group 的默认值为 0。
- span([group])：返回(start(group), end(group))。

Match 对象的相关属性和方法示例如下：

```
import re
t="19:05:25"
m=re.match(r'^(\d\d)\:(\d\d)\:(\d\d)$', t)     #r 原义
print("m.string:", m.string)                  #m.string: 19:05:25
print(m.re)        #re.compile('^(\\d\\d)\\:(\\d\\d)\\:((\\d\\d))$')
print("m.pos:", m.pos)                        #m.pos: 0
print("m.endpos:", m.endpos)                  #m.endpos: 8
print("m.lastindex:", m.lastindex)            #m.lastindex: 3
print("m.lastgroup:", m.lastgroup)            #m.lastgroup: None
print("m.group(0):", m.group(0))              #m.group(0): 19:05:25
print("m.group(1,2):", m.group(1, 2))         #m.group(1,2):('19', '05')
print("m.groups():", m.groups())              #m.groups():('19', '05', '25')
print("m.groupdict():", m.groupdict())        #m.groupdict(): {}
print("m.start(2):", m.start(2))              #m.start(2): 3
print("m.end(2):", m.end(2))                  #m.end(2): 5
print("m.span(2):", m.span(2))                #m.span(2):(3, 5)
```

关于分组的内容见后面。

2）分组

除了简单地判断是否匹配之外，正则表达式还有提取子串的强大功能，用()表示的就是要提取的分组。例如^(\d{3})-(\d{3,8})$分别定义了两个组，可以直接从匹配的字符串中提取出区号和本地号码：

```
>>> m=re.match(r'^(\d{3})-(\d{3,8})$', '010-12345')
>>> m.group(0)      #'010-12345'
>>> m.group(1)      #'010'
>>> m.group(2)      #'12345'
```

如果正则表达式中定义了组，就可以在 Match 对象上用 group()方法提取出子串。注意 group(0)永远是原始字符串，group(1)、group(2)表示第 1、2 个子串。

3）切分字符串

用正则表达式切分字符串比用固定的字符更灵活，请看普通字符串的切分代码：

```
>>> 'a b   c'.split(' ')                         #split(' ')表示按空格分隔
['a', 'b', '', '', 'c']
```

其结果是无法识别连续的空格，可以使用 re.split()方法来分割字符串，例如 re.split(r'\s+', text)将字符串按空格分割成一个单词列表：

```
>>> re.split(r'\s+', 'a b   c')                  #用正则表达式
['a', 'b', 'c']
```

无论多少个空格都可以正常分割。

再例如分隔符既有空格又有逗号、分号的情况：

```
>>> re.split(r'[\s\,]+', 'a,b, c  d')           #可以识别空格、逗号
['a', 'b', 'c', 'd']
>>> re.split(r'[\s\,\;]+', 'a,b;; c  d')        #可以识别空格、逗号、分号
['a', 'b', 'c', 'd']
```

4）search()和 findall()方法

re.match()总是从字符串“开头”去匹配，并返回匹配的字符串的 Match 对象，所以当用 re.match()去匹配非“开头”部分的字符串时会返回 None。

```
str1='Hello World!'
print(re.match(r'World',str1))                   #结果为 None
```

如果想在字符串中的任意位置去匹配，请用 re.search()或 re.findall()。

re.search()将对整个字符串进行搜索，并返回第 1 个匹配的字符串的 Match 对象。

```
str1='Hello World!'
print(re.search(r'World',str1))
```

输出结果如下：

```
<_sre.SRE_Match object; span=(6,11), match='World'>
```

re.findall()函数将返回一个所有匹配的字符串的字符串列表。例如：

```
str1='Hi, I am Shirley Hilton. I am his wife.'
>>> print(re.search(r'hi',str1))
```

以上代码的输出结果如下：

```
<_sre.SRE_Match object; span=(10, 12), match='hi'>
```

此时应用 re.findall()函数：

```
>>> re.findall(r'hi',str1)
```

输出结果如下：

```
['hi', 'hi']
```

这两个“hi”分别来自“Shirley”和“his”。在默认情况下正则表达式是严格区分大小写的，所以“Hi”和“Hilton”中的“Hi”被忽略了。

如果只想找到“hi”这个单词，而不把包含它的单词计算在内，那就可以使用“\bhi\b”这个正则表达式。“\b”在正则表达式中表示单词的开头或结尾，空格、标点、换行都算是单词的分割；而“\b”自身又不会匹配任何字符，它代表的只是一个位置，所以单词前后的空格、标点之类不会出现在结果中。

在前面的例子中，“\bhi\b”匹配不到任何结果，因为没有单词 hi（“Hi”不是，严格区分大小写）。但如果是“\bhi”，就可以匹配到 1 个“hi”，出自“his”。

5.3.2　中文分词

在英文中，单词之间是以空格作为自然分界符的；而中文只是句子和段可以通过明显的分界符来简单划分，唯独词没有一个形式上的分界符，虽然也同样存在短语之间的划分问题，但是在词这一层上，中文要比英文复杂得多。

中文分词就是将连续的字序列按照一定的规范重新组合成词序列的过程。中文分词是网页分析索引的基础。分词的准确性对搜索引擎来说十分重要，如果分词速度太慢，即使再准确，对于搜索引擎来说也是不可用的，因为搜索引擎需要处理很多网页，如果分析消耗的时间过长，会严重影响搜索引擎内容更新的速度。因此，搜索引擎对于分词的准确率和速率都提出了很高的要求。

jieba 是一个支持中文分词、高准确率、高效率的 Python 中文分词组件，它支持繁体分词和自定义词典，并支持 3 种分词模式。

（1）精确模式：试图将句子最精确地切开，适合文本分析。

（2）全模式：把句子中所有可以成词的词语都扫描出来，速度非常快，但是不能解决歧义的问题。

（3）搜索引擎模式：在精确模式的基础上对长词再次切分，提高召回率，适合用于搜索引擎分词。

5.3.3　安装和使用 jieba

在命令行中输入以下代码：

```
pip install  jieba
```

如果出现以下提示则安装成功。

```
Installing collected packages: jieba
  Running setup.py install for jieba ... done
Successfully installed jieba-0.38
```

组件提供了 jieba.cut()方法用于分词，cut()方法接受两个输入参数：

（1）第 1 个参数为需要分词的字符串。

（2）cut_all 参数用来控制分词模式。

jieba.cut()返回的结构是一个可迭代的生成器（generator），可以使用 for 循环来获得分词后得到的每一个词语，也可以用 list(jieba.cut(...))转化为 list 列表。例如：

```
import jieba
seg_list=jieba.cut("我来到北京清华大学", cut_all=True)  #全模式
print("Full Mode:", '/'.join(seg_list))
seg_list=jieba.cut("我来到北京清华大学")  #默认是精确模式，或者 cut_all =False
print(type(seg_list))                                  #<class 'generator'>
print("Default Mode:", '/'.join(seg_list))
seg_list=jieba.cut_for_search("我来到北京清华大学")  #搜索引擎模式
print("搜索引擎模式:", '/'.join(seg_list))
seg_list=jieba.cut("我来到北京清华大学")
for word in seg_list:
   print(word,end=' ')
```

运行结果如下：

```
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\ADMINI~1\AppData\Local\Temp\jieba.cache
Loading model cost 1.648 seconds.
Prefix dict has been built succesfully.
Full Mode: 我/来到/北京/清华/清华大学/华大/大学
<class 'generator'>
Default Mode: 我/来到/北京/清华大学
搜索引擎模式: 我/来到/北京/清华/华大/大学/清华大学
我 来到 北京 清华大学
```

jieba.cut_for_search()方法仅有一个参数，为分词的字符串，该方法适用于搜索引擎构造倒排索引的分词，粒度比较细。

5.3.4　为 jieba 添加自定义词典

国家 5A 级景区存在很多与旅游相关的专有名词，举个例子：

[输入文本] 故宫的著名景点包括乾清宫、太和殿和黄琉璃瓦等

[精确模式] 故宫/的/著名景点/包括/乾/清宫/、/太和殿/和/黄/琉璃瓦/等

[全模式]故宫/的/著名/著名景点/景点/包括/乾/清宫/太和/太和殿/和/黄/琉璃/琉璃瓦/等

显然，专有名词乾清宫、太和殿、黄琉璃瓦（假设为一个文物）可能因分词而分开，这也是很多分词工具的一个缺陷。但是 jieba 分词支持开发者使用自定义的词典，以便包含 jieba 词库里没有的词语。虽然 jieba 有新词识别能力，但自行添加新词可以保证更高的正确率，尤其是专有名词。

其基本用法如下：

```
jieba.load_userdict(file_name)      #file_name 为自定义词典的路径
```

词典格式是一个词占一行；每一行分 3 个部分，一部分为词语，另一部分为词频，最后为词性（可省略，jieba 的词性标注方式和 ICTCLAS 的标注方式一样。ns 为地点名词，nz 为其他专用名词，a 是形容词，v 是动词，d 是副词），3 个部分用空格隔开。例如以下自定义词典 dict.txt：

```
乾清宫 5 ns
黄琉璃瓦 4
云计算 5
李小福 2 nr
八一双鹿 3 nz
凯特琳 2 nz
```

下面是导入自定义词典后再分词。

```
import jieba
jieba.load_userdict("dict.txt")                    #导入自定义词典
text="故宫的著名景点包括乾清宫、太和殿和黄琉璃瓦等"
seg_list=jieba.cut(text, cut_all=False)     #精确模式
print("[精确模式]: ", "/ ".join(seg_list))
```

输出结果如下，其中专有名词连在一起，即乾清宫和黄琉璃瓦。

```
[精确模式]:故宫/ 的/ 著名景点/ 包括/ 乾清宫/ 、/ 太和殿/ 和/ 黄琉璃瓦/ 等
```

5.3.5　文本分类的关键词提取

当文本分类时，在构建 VSM（向量空间模型）的过程中或者把文本转换成数学形式的计算中，需要运用到关键词提取的技术，jieba 可以简便地提取关键词。

其基本用法如下：

```
jieba.analyse.extract_tags(sentence, topK=20, withWeight =False, allowPOS=())
```

这里需要先 import jieba.analyse。其中，sentence 为待提取的文本；topK 为返回几个 TF/IDF 权重最大的关键词，默认值为 20；withWeight 为是否一并返回关键词权重值，默认值为 False；allowPOS 指仅包含指定词性的词，默认值为空，即不进行筛选。

```
import jieba,jieba.analyse
jieba.load_userdict("dict.txt")                        #导入自定义词典
```

```
text="故宫的著名景点包括乾清宫、太和殿和午门等。其中乾清宫非常精美，午门是紫禁城的正门，午门居中向阳。"
seg list=jieba.cut(text, cut all=False)
print("分词结果：", "/".join(seg_list))                #精确模式
tags=jieba.analyse.extract tags(text, topK=5)          #获取关键词
print("关键词：", " ".join(tags))
tags=jieba.analyse.extract tags(text, topK=5,withWeight=True)
                                                        #返回关键词权重值
print(tags)
```

输出结果如下：

```
分词结果： 故宫/的/著名景点/包括/乾清宫/、/太和殿/和/午门/等/。/其中/乾清宫/非常/精美/，/午门/是/紫禁城/的/正门/，/午门/居中/向阳/。
关键词： 午门 乾清宫 著名景点 太和殿 向阳
[('午门', 1.5925323525975001),('乾清宫', 1.4943459378625),('著名景点', 0.86879235325),('太和殿', 0.63518800210625),('向阳', 0.578517922051875)]
```

其中，午门出现 3 次、乾清宫出现两次、著名景点出现 1 次。如果 topK=5，按照顺序输出提取 5 个关键词，则输出“午门 乾清宫 著名景点 太和殿 向阳”。

```
jieba.analyse.TFIDF(idf_path=None)#新建 TF/IDF 实例，idf_path 为 IDF 频率文件
```

关键词提取所使用的逆向文件频率（IDF）文本语料库可以切换成自定义语料库的路径。

```
jieba.analyse.set_idf_path(file_name)     #file_name 为自定义语料库的路径
```

关键词提取所使用的停止词（Stop Words）文本语料库可以切换成自定义语料库的路径。

说明：TF/IDF 是一种统计方法，用于评估一字词对于一个文件集或一个语料库中的一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。TF/IDF 的主要思想是：如果某个词或短语在一篇文章中出现的频率 TF 高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

5.3.6 deque

deque（double-ended queue 的缩写）双向队列类似于 list 列表，位于 Python 标准库 collections 中。它提供了两端都可以操作的序列，这意味着在序列的前后都可以执行添加或删除操作。

❶ 创建双向队列 deque

```
from collections import deque
d=deque()
```

❷ 添加元素

```
d=deque()
```

```
d.append(3)
d.append(8)
d.append(1)
```

那么此时 d=deque([3,8,1])、len(d)=3、d[0]=3、d[−1]=1。

deque 支持从任意一端添加元素。append()用于从右端添加一个元素，appendleft()用于从左端添加一个元素。

❸ 两端都使用 pop

```
d=deque(['1', '2', '3', '4', '5'])
```

d.pop()抛出的是'5'，d.popleft()抛出的是'1'，可见默认 pop()抛出的是最后一个元素。

❹ 限制 deque 的长度

```
d=deque(maxlen=20)
for i in range(30):
    d.append(str(i))
```

此时 d 的值为 d=deque(['10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20', '21', '22', '23', '24', '25', '26', '27', '28', '29'], maxlen=20)，可见当限制长度的 deque 增加超过限制数的项时另一边的项会自动删除。

❺ 添加 list 的各项到 deque 中

```
d=deque([1,2,3,4,5])
d.extend([0])
```

那么此时 d=deque([1,2,3,4,5,0])。

```
d.extendleft([6,7,8])
```

此时 d=deque([8,7,6,1,2,3,4,5,0])。

5.4 程序设计的步骤

5.4.1 信息采集模块——网络爬虫的实现

视频讲解

网络爬虫的实现原理及过程如下：

（1）获取初始的 URL。初始的 URL 地址可以由用户指定的某个或某几个初始爬取网页决定。

（2）根据初始的 URL 爬取页面并获得新的 URL。在获得初始的 URL 地址之后，首先需要爬取对应 URL 地址中的网页，在爬取了对应的 URL 地址中的网页后将网页存储到原始数据库中，并且在爬取网页的同时发现新的 URL 地址，将已爬取的 URL 地址存放到一个已爬 URL 列表中，用于去重复判断爬取的进程。

（3）将新的 URL 放到 URL 队列中。注意，在第 2 步中获取了下一个新的 URL 地址

之后会将新的 URL 地址放到 URL 队列中。

（4）从 URL 队列中读取新的 URL，并依据新的 URL 爬取网页，同时从新网页中获取新 URL，并重复上述的爬取过程。

（5）当满足爬虫系统设置的停止条件时停止爬取。在编写爬虫的时候一般会设置相应的停止条件，如果没有设置停止条件，爬虫会一直爬取下去，直到无法获取新的 URL 地址为止，若设置了停止条件，爬虫则会在停止条件满足时停止爬取。

根据图 5-4 所示的网络爬虫的实现原理及过程，这里指定中原工学院新闻门户的 URL 地址“http://www.zut.edu.cn/index/xwdt.htm”为初始的 URL。

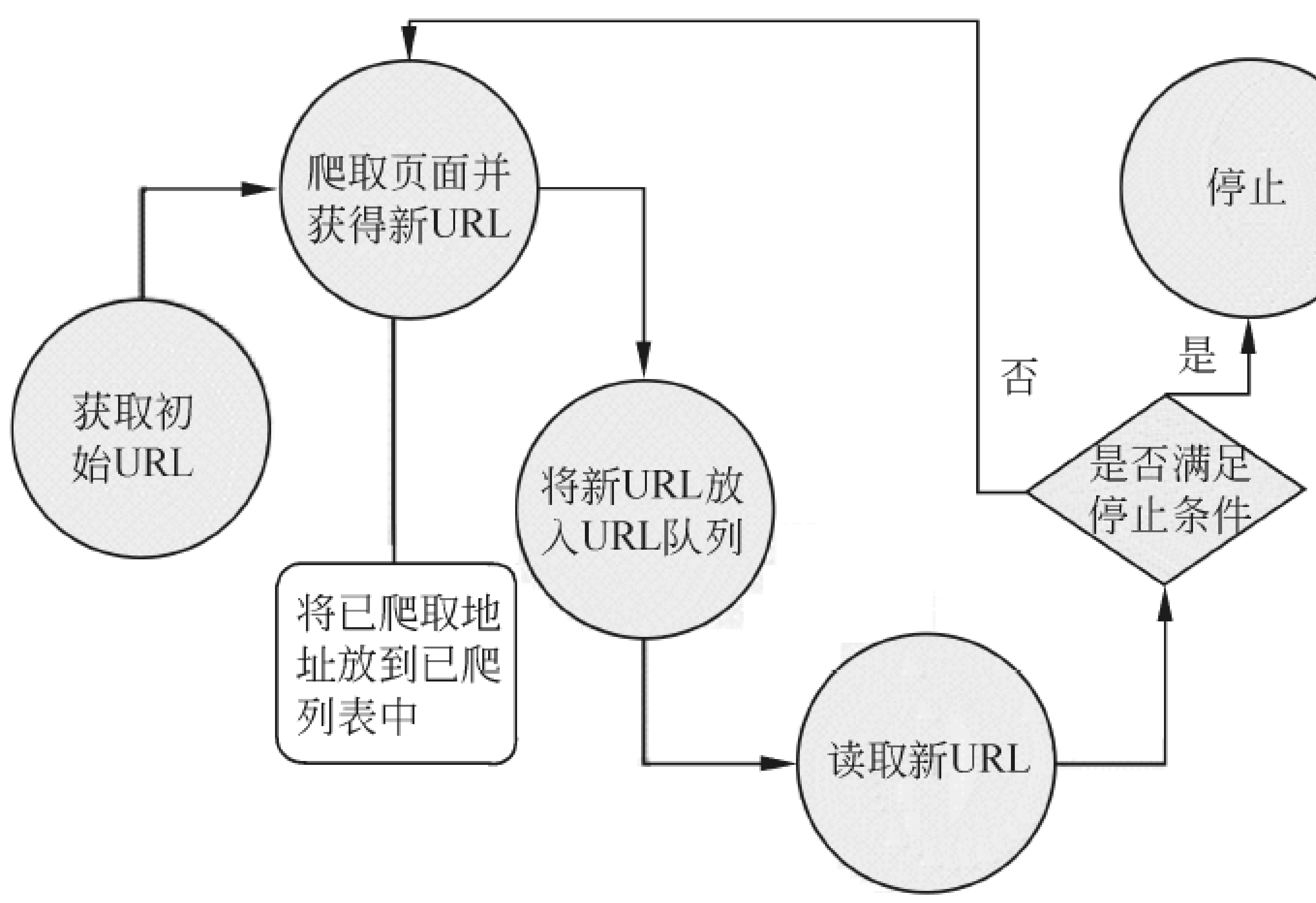


图 5-4　网络爬虫的实现原理及过程

使用 unvisited 队列存储待爬取 URL 链接的集合并使用广度优先搜索，使用 visited 集合存储已访问过的 URL 链接。

```
unvisited=deque()        #待爬取链接的列表，使用广度优先搜索
visited=set()            #已访问的链接集合
```

在数据库中建立两个 table，其中一个是 doc 表，存储每个网页 ID 和 URL 链接。

```
create table doc(id int primary key,link text)
```

例如：

```
1   http://www.zut.edu.cn/index/xwdt.htm
2   http://www.zut.edu.cn/info/1052/19838.htm
3   http://www.zut.edu.cn/info/1052/19837.htm
4   http://www.zut.edu.cn/info/1052/19836.htm
5   http://www.zut.edu.cn/info/1052/19835.htm
6   http://www.zut.edu.cn/info/1052/19834.htm
7   http://www.zut.edu.cn/info/1052/19833.htm
……
```

另一个是 word 表，即倒排表，存储词语和其对应的网页 ID 的 list。

```
create table word(term varchar(25) primary key,list text)
```

如果一个词在某个网页中出现多次，那么 list 中这个网页的序号也出现多次。list 最后转换成一个字符串存进数据库。

例如，词“王宗敏”出现在网页 ID 为 12、35、88 号的网页里，12 号页面 1 次，35 号页面 3 次，88 号页面两次，它的 list 应为[12,35,35,35,88,88]，转换成字符串"12 35 35 35 88 88"存储在 word 表的一条记录中，形式如下：

term	list
王宗敏	12　35　35　35　88　88
校友会	54　190　190　701　986　986　1024

爬取中原工学院新闻网页的代码如下：

```
#search engine build-2.py
import sys
from collections import deque
import urllib
from urllib import request
import re
from bs4 import BeautifulSoup
import lxml
import sqlite3
import jieba

url='http://www.zut.edu.cn'    #入口

unvisited=deque()              #待爬取链接的集合，使用广度优先搜索
visited=set()                  #已访问的链接集合
unvisited.append(url)

conn=sqlite3.connect('viewsdu.db')
c=conn.cursor()
#在 create table 之前先 drop table 是因为之前测试的时候已经建过 table 了，所以再次运
#行代码的时候要把旧的 table 删了重建
c.execute('drop table doc')
c.execute('create table doc(id int primary key,link text)')
c.execute('drop table word')
c.execute('create table word(term varchar(25) primary key,list text)')
conn.commit()
conn.close()
print('***************开始爬取*************************')
cnt=0
print('开始。。。。。 ')
while unvisited:
    url=unvisited.popleft()
```

```
    visited.add(url)
    cnt+=1
    print('开始抓取第',cnt,'个链接：',url)

    #爬取网页内容
    try:
        response=request.urlopen(url)
        content=response.read().decode('utf-8')

    except:
        continue
    #寻找下一个可爬的链接，因为搜索范围是网站内，所以对链接有格式要求，需根据具体情况而定
    #解析网页内容，可能有几种情况，这也是根据这个网站网页的具体情况写的
    soup=BeautifulSoup(content,'lxml')
    all_a=soup.find_all('a',{'class':"c67214"})   #本页面所有的新闻链接<a>
    for a in all_a:
        #print(a.attrs['href'])
        x=a.attrs['href']                         #网址
        if re.match(r'http.+',x):#排除是http开头，而不是http://www.zut.edu.cn网址
            if not re.match(r'http\:\/\/www\.zut\.edu\.cn\/.+',x):
                continue
        if re.match(r'\/info\/.+',x):             #"/info/1046/20314.htm"
            x='http://www.zut.edu.cn'+x
        elif re.match(r'info/.+',x):              #"info/1046/20314.htm"
            x='http://www.zut.edu.cn/'+x
        elif re.match(r'\.\.\/info/.+',x):        #"../info/1046/20314.htm"
            x='http://www.zut.edu.cn'+x[2:]
        elif re.match(r'\.\.\/\.\.\/info/.+',x): #"../../info/1046/20314.htm"
            x='http://www.zut.edu.cn'+x[5:]
        #print(x)
        if(x not in visited) and(x not in unvisited):
                unvisited.append(x)

    a=soup.find('a',{'class':"Next"})             #下一页<a>
    if a!=None:
        x=a.attrs['href']                         #网址
        if re.match(r'xwdt\/.+',x):
            x='http://www.zut.edu.cn/index/'+x
        else:
            x='http://www.zut.edu.cn/index/xwdt/'+x
        if(x not in visited) and(x not in unvisited):
           unvisited.append(x)
```

以上代码实现要爬取的网址队列 unvisited。

5.4.2　索引模块——建立倒排词表

解析新闻网页内容，这个过程需要根据这个网站网页的具体情况来处理。

```
soup=BeautifulSoup(content,'lxml')
title=soup.title
article=soup.find('div',class ='c67215 content',id='vsb newscontent')
author=soup.find('span',class ="authorstyle67215")  #作者
time=soup.find('span',class ="timestyle67215")
if title==None and article==None and author==None:
    print('无内容的页面。')
    continue
elif article==None and author==None:
    print('只有标题。')
    title=title.text
    title=''.join(title.split())
    article=''
    author=''
elif article==None:
    print('有标题有作者，缺失内容')
    title=title.text
    title=''.join(title.split())
    article=''
    author=author.get text("",strip=True)
    author=''.join(author.split())
elif author==None:
    print('有标题有内容，缺失作者')
    title=title.text
    title=''.join(title.split())
    article=article.get text("",strip=True)
    article=''.join(article.split())
    author=''
else:
    title=title.text
    title=''.join(title.split())
    article=article.get text("",strip=True)
    article=''.join(article.split())
    author=author.get text("",strip=True)
    author=''.join(author.split())
print('网页标题：',title)
```

提取出的网页内容存在于 title、article、author 中，对它们进行中文分词，并对每个分出的词语建立倒排词表。

```
    seggen=jieba.cut_for_search(title)
```

```
    seglist=list(seggen)
    seggen=jieba.cut for search(article)
    seglist+=list(seggen)
    seggen=jieba.cut for search(author)
    seglist+=list(seggen)

    #数据存储
    conn=sqlite3.connect("viewsdu.db")
    c=conn.cursor()
    c.execute('insert into doc values(?,?)',(cnt,url))
    #对每个分出的词语建立倒排词表
    for word in seglist:
        #print(word)
        #检验看看这个词语是否已存在于数据库
        c.execute('select list from word where term=?',(word,))
        result=c.fetchall()
        #如果不存在
        if len(result)==0:
            docliststr=str(cnt)
            c.execute('insert into word values(?,?)',(word,docliststr))
        #如果已存在
        else:
            docliststr=result[0][0]        #得到字符串
            docliststr+=' '+str(cnt)
            c.execute('update word set list=? where term=?',(docliststr,word))
    conn.commit()
    conn.close()
print('词表建立完毕!! ')
```

以上代码只需运行一次即可，搜索引擎所需的数据库已经建好。运行上述代码出现如下结果：

```
开始抓取第 110 个链接： http://www.zut.edu.cn/info/1041/20191.htm
网页标题： 我校 2017 年学生奖助项目评审工作完成资助育人成效显著-中原工学院
开始抓取第 111 个链接： http://www.zut.edu.cn/info/1041/20190.htm
网页标题： 我校教师李慕杰、王学鹏参加中国致公党河南省第一次代表大会-中原工学院
开始抓取第 112 个链接： http://www.zut.edu.cn/info/1041/20187.htm
网页标题： 我校与励展企业开展校企合作-中原工学院
开始抓取第 113 个链接： http://www.zut.edu.cn/info/1041/20184.htm
网页标题： 平顶山学院李培副校长一行来我校考察交流-中原工学院
开始抓取第 114 个链接： http://www.zut.edu.cn/info/1041/20179.htm
网页标题： 我校学生在工程造价技能大赛中获佳绩-中原工学院
开始抓取第 115 个链接： http://www.zut.edu.cn/info/1041/20178.htm
网页标题： 我校召开 2018 届毕业生就业工作会议-中原工学院
```

5.4.3　网页排名和搜索模块

视频讲解

当需要搜索的时候执行 search_engine_use.py，完成网页排名和搜索功能。

网页排名采用 TF/IDF 统计。TF/IDF 是一种用于信息检索与数据挖掘的常用加权技术。TF/IDF 统计用于评估一词对于一个文件集或一个语料库中的一份文件的重要程度。TF 的意思是词频（Term Frequency），IDF 的意思是逆文本频率指数（Inverse Document Frequency）。TF 表示词条 t 在文档 d 中出现的频率。IDF 的主要思想是：如果包含词条 t 的文档越少，则词条 t 的 IDF 越大，说明词条 t 具有很好的类别区分能力。

词条 t 的 IDF 计算公式如下：

$$idf= \log(N/df)$$

其中，N 是文档总数，df 是包含词条 t 的文档数量。

在本程序中 tf={文档号：出现次数}存储的是某个词在文档中出现的次数。例如王宗敏的 tf={12:1，35:3，88:2}即词“王宗敏”出现在网页 ID 为 12、35、88 号的网页里，12 号页面 1 次，35 号页面 3 次，88 号页面两次。

score={文档号：文档得分}用于存储命中（搜到）文档的排名得分。

```
#search engine use.py
import re
import urllib
from urllib import request
from collections import deque
from bs4 import BeautifulSoup
import lxml
import sqlite3
import jieba
import math
conn=sqlite3.connect("viewsdu.db")
c=conn.cursor()
c.execute('select count(*) from doc')
N=1+c.fetchall()[0][0]                          #文档总数
target=input('请输入搜索词：')
seggen=jieba.cut_for_search(target)             #将搜索内容分词
score={}                                        #字典，用于存储“文档号：文档得分”
for word in seggen:
    print('得到查询词：',word)
    tf={}                                       #文档号：次数{12：1，35：3，88：2}
    c.execute('select list from word where term=?',(word,))
    result=c.fetchall()
    if len(result)>0:
        doclist=result[0][0]                    #字符串“12 35 35 35 88 88”
        doclist=doclist.split(' ')
        doclist=[int(x) for x in doclist] #['12','35','35','35','88','88']
```

```
                                     #把字符串转换成元素为 int 的 list[12,35,88]
        df=len(set(doclist)) #当前 word 对应的 df 数，注意 set 集合实现去掉重复项
        idf=math.log(N/df)   #计算出 IDF
        print('idf: ',idf)
        for num in doclist:  #计算词频 TF，即在某文档中出现的次数
            if num in tf:
                tf[num]=tf[num]+1
            else:
                tf[num]=1
        #TF 统计结束，现在开始计算 score
        for num in tf:
            if num in score:
                #如果该 num 文档已经有分数了，则累加
                score[num]=score[num]+tf[num]*idf
            else:
                score[num]=tf[num]*idf
sortedlist=sorted(score.items(),key=lambda d:d[1],reverse=True)
                                                    #对 score 字典按字典的值排序
#print('得分列表',sortedlist)
cnt=0
for num,docscore in sortedlist:
    cnt=cnt+1
    c.execute('select link from doc where id=?',(num,))
                                                    #按照 ID 获取文档的连接（网址）
    url=c.fetchall()[0][0]
    print(url ,'得分：',docscore)                    #输出网址和对应得分
    try:
        response=request.urlopen(url)
        content=response.read().decode('utf-8') #可以输出网页内容
    except:
        print('oops...读取网页出错')
        continue
    #解析网页输出标题
    soup=BeautifulSoup(content,'lxml')
    title=soup.title
    if title==None:
        print('No title.')
    else:
        title=title.text
        print(title)
    if cnt>20:                                  #超过 20 条则结束，即输出前 20 条
        break
if cnt==0:
    print('无搜索结果')
```

当运行 search_engine_use.py 时出现如下提示：

```
请输入搜索词：王宗敏
Building prefix dict from the default dictionary ...
Loading model from cache C:\Users\xmj\AppData\Local\Temp\jieba.cache
Loading model cost 0.961 seconds.
Prefix dict has been built succesfully.
得到查询词：王宗敏
idf: 3.337509562404897
http://www.zut.edu.cn/info/1041/20120.htm 得分：13.350038249619589
王宗敏校长一行参加深圳校友会年会并走访合作企业-中原工学院
http://www.zut.edu.cn/info/1041/20435.htm 得分：13.350038249619589
中国工程院张彦仲院士莅临我校指导工作-中原工学院
http://www.zut.edu.cn/info/1041/19775.htm 得分：10.012528687214692
我校河南省功能性纺织材料重点实验室接受现场评估-中原工学院
http://www.zut.edu.cn/info/1041/19756.htm 得分：10.012528687214692
王宗敏校长召开会议推进“十三五”规划“八项工程”建设-中原工学院
http://www.zut.edu.cn/info/1041/19726.htm 得分：10.012528687214692
我校 2017 级新生开学典礼隆重举行-中原工学院
```

第 6 章

爬虫应用——抓取百度图片

视频讲解

6.1　程序功能介绍

使用网络爬虫技术爬取百度图片某主题的相关图片，并且能按某一关键字搜索图片下载到本地指定的文件夹中。本程序主要完成下载功能，不需要设计图形化界面。在运行时出现如下提示：

```
Please input you want search:
```

让用户输入关键词，例如输入“夏敏捷”，然后按回车键，则看到如图 6-1 所示的效果。

```
======= RESTART: I:\（第14本） Python轻松学项目案例开发 （Python课程设计案例）\百度图片下载00.py =======
Please input you want search: 夏敏捷
Current page:0**********************************
开始下载:http://img14.360buyimg.com/n0/jfs/t1273/309/228568970/10584/3d71a1e0/550b853bN692a3537.jpg
开始下载:http://img13.360buyimg.com/n0/jfs/t586/241/26929280/71476/2c65610c/54484fe6Nb33010bd.jpg
开始下载:http://img002.21cnimg.com/photos/album/20160511/m600/5C2557BC2AEC044F4324EAEEBCAF018C.jpeg
开始下载:http://img3x2.ddimg.cn/10/9/1293525892-1_u_2.jpg
开始下载:http://imgs.soufun.com/news/2014_08/01/news/1406859161065_000.jpg
开始下载:http://g-ec4.images-amazon.com/images/G/28/BOOK-Catalog/shfz/20101117/B0040IF6VQ_01_amzn.jpg
开始下载:http://imgs.soufun.com/news/2013_09/30/news/1380523396671_000.jpg
开始下载:http://img3m1.ddimg.cn/39/22/1067418831-1_u_1.jpg
开始下载:http://qnimg.zowoyoo.com/img/280179/1493550929586.jpg
开始下载:http://img1.gtimg.com/house_guangzhou/pics/hv1/75/165/1475/95954025.jpg
```

图 6-1　爬取百度图片运行效果示意图

从图 6-1 可以看到开始下载了。

6.2　程序设计的思路

一般来说，制作一个爬虫需要分以下几个步骤：

（1）分析需求，这里的需求就是爬取网页图片。

（2）分析网页源代码和网页结构，配合 F12 键查看网页源代码。

（3）编写正则表达式或者 XPath 表达式。

（4）正式编写 Python 爬虫代码。

本章按照该步骤实现按关键词爬取百度图片。

6.3　关键技术

6.3.1　图片文件下载到本地

❶ 使用 request.urlretrieve()函数

如果要把对应图片文件下载到本地，可以使用 urlretrieve()函数。

```
from urllib import request
request.urlretrieve("http://www.zzti.edu.cn/ mediafile/index/2017/06/24
/1qjdyc7vq5.jpg","aaa.jpg")
```

上例就可以把网络上中原工学院的图片资源 1qjdyc7vq5.jpg 下载到本地，生成 aaa.jpg 图片文件。

❷ 使用 Python 的文件操作函数 write()写入文件

```
from urllib import request
import urllib
url=' http://www.zzti.edu.cn/ mediafile/index/2017/06/24/1qjdyc7vq5.jpg '
url1=urllib.request.Request(url)           #Request()函数将 url 添加到头部，
                                           #模拟浏览器访问
page=urllib.request.urlopen(url1).read()   #将 url 页面的源代码保存成字符串
#open().write()方法原始且有效
open('C:\\aaa.jpg', 'wb').write(page)      #写入 aaa.jpg 文件中
```

6.3.2　爬取指定网页中的图片

首先用 urllib 库来模拟浏览器访问网站的行为，由给定的网站链接（url）得到对应网页的源代码（html 标签）。其中，源代码以字符串的形式返回。

然后用正则表达式 re 库在字符串（网页源代码）中匹配表示图片链接的小字符串，返回一个列表。

最后循环列表，根据图片链接将图片保存到本地。

urllib 库的使用在 Python2.x 和 Python3.x 中的差别很大，本案例以 Python3.x 为例。

```
'''
    第一个简单的爬取图片程序，使用 Python3.x 和 urllib 与 re 库
'''
import urllib.request
```

```
import re                  #正则表达式
def getHtmlCode(url):   #该方法传入 url，返回 url 的 html 的源代码
    headers={
        'User-Agent': 'Mozilla/5.0(Linux; Android 6.0; Nexus 5 Build/MRA58N)
        AppleWebKit/537.36(KHTML, like Gecko) Chrome/56.0.2924.87 Mobile
        Safari/537.36'
    }
    url1=urllib.request.Request(url, headers=headers)
                                  #Request()函数将 url 添加到头部，模拟浏览器访问
    page=urllib.request.urlopen(url1).read()
                                  #将 url 页面的源代码保存成字符串
    page=page.decode('UTF-8')     #字符串转码
    return page

def getImg(page):#该方法传入 html 的源代码，经过截取其中的 img 标签，将图片保存到本机
    imgList=re.findall(r'(http:[^\s]*?(jpg|png|gif))"',page)
    x=0
    for imgUrl in imgList:        #列表循环
        try:
            print('正在下载：%s'%imgUrl[0])
            #urlretrieve(url,local)方法根据图片的 url 将图片保存到本机
            urllib.request.urlretrieve(imgUrl[0],'E:/img/%d.jpg'%x)
            x+=1
        except:
             continue

if __name__=='__main__':
    url='http://blog.csdn.net/qq 32166627/article/details/60345731'
                                  #指定网址页面
    page=getHtmlCode(url)
    getImg(page)
```

对于 findall(正则表达式,代表页面源代码的 str)函数，在字符串中按照正则表达式截取其中的子字符串，findall()返回一个列表，列表中的元素是一个个元组，元组的第 1 个元素是图片的 url，第 2 个元素是 url 的扩展名，列表形式如下：

```
[('http://avatar.csdn.net/4/E/B/1 qq 32166627.jpg', 'jpg'),
('http://avatar.csdn.net/1/1/4/2 fly yr.jpg', 'jpg'),
('http://avatar.csdn.net/8/1/3/2 u013007900.jpg', 'jpg'),
…
('http://avatar.csdn.net/1/B/B/1_csdn.jpg', 'jpg')]
```

上述代码在找图片的 url 时用的是 re（正则表达式）。re 用得好会有奇效，用得不好效果极差。

既然得到了网页的源代码，就可以根据标签的名称得到其中的内容。

由于正则表达式难以掌握，这里用一个第三方库——BeautifulSoup，它可以根据标签

的名称对网页内容进行截取。BeautifulSoup4 的中文文档请参见页面“http://beautifulsoup.readthedocs.io/zh_CN/latest/”。

6.3.3　BeautifulSoup 库概述

BeautifulSoup（英文原意是美丽的蝴蝶）是一个 Python 处理 HTML/XML 的函数库，是 Python 内置的网页分析工具，用来快速地转换被抓取的网页。它产生一个转换后 DOM 树，尽可能和原文档内容的含义一致，这种措施通常能够满足用户搜集数据的需求。

BeautifulSoup 提供了一些简单的方法以及类 Python 语法来查找、定位、修改一棵转换后 DOM 树。BeautifulSoup 自动将送进来的文档转换为 Unicode 编码，而且在输出的时候转换为 UTF-8。BeautifulSoup 可以找出“所有的链接<a>”，或者“所有 class 是 xxx 的链接<a>”，再或者是“所有匹配.cn 的链接 url”。

❶ BeautifulSoup 的安装

使用 pip 直接安装 BeautifulSoup4：

```
pip3 install beautifulsoup4
```

推荐在现在的项目中使用 BeautifulSoup4（bs4），导入时需要 import bs4。

❷ BeautifulSoup 的基本使用方式

下面使用一段代码演示 BeautifulSoup 的基本使用方式。

```
from bs4 import BeautifulSoup
#doc 可以是一个 HTML 内容的字符串，本例是列表，需要转换成字符串
doc=['<html><head><title> The story of Monkey </title></head>',
     '<body><p id="firstpara" align="center">This is one paragraph </p>',
     '<p id="secondpara" align="center">This is two paragraph </p>',
     '</html>']
soup=BeautifulSoup(''.join(doc), "html.parser")
                              #提供字符串信息，''.join(doc)将其合并为字符串
print(soup.prettify())
```

在使用时 BeautifulSoup 首先要导入 bs4 库：

```
from bs4 import BeautifulSoup
```

创建 BeautifulSoup 对象：

```
soup=BeautifulSoup(html)
```

另外，还可以用本地 HTML 文件来创建对象，例如：

```
soup=BeautifulSoup(open('index.html') , "html.parser")#提供本地 HTML 文件
```

上面的代码是将本地 index.html 文件打开，用它来创建 soup 对象。

用户也可以使用网址 URL 获取 HTML 文件，例如：

```
from urllib import request
response=request.urlopen("http://www.baidu.com")
```

```
html=response.read()
html=html.decode("utf-8")  #decode()用于将网页的信息进行解码，否则会产生乱码
soup=BeautifulSoup(html, "html.parser")          #远程网站上的 HTML 文件
```

程序段最后格式化输出 BeautifulSoup 对象的内容。

```
print(soup.prettify())
```

运行结果如下：

```
<html>
 <head>
  <title> The story of Monkey </title>
 </head>
 <body>
  <p align="center" id="firstpara">
   This is one paragraph
   </p>
   <p align="center" id="secondpara">
    This is two paragraph
   </p>
 </body>
</html>
```

以上便是输出结果，格式化打印出了 BeautifulSoup 对象（DOM 树）的内容。

BeautifulSoup 将复杂的 HTML 文档转换成一个复杂的树形结构，其中每个结点都是 Python 对象。所有对象可以归纳为 4 种，即 Tag、NavigableString、BeautifulSoup（前面例子中已经使用过）、Comment。

1）Tag 对象

Tag 是什么？通俗点讲就是 HTML 中的一个个标签，例如：

```
<title> The story of Monkey </title>
<a href="http://example.com/elsie" id="link1">Elsie</a>
```

上面的<title>、<a>等 HTML 标签加上里面包括的内容就是 Tag，下面用 BeautifulSoup 来获取 Tags。

```
print(soup.title)
print(soup.head)
```

输出如下：

```
<title> The story of Monkey </title>
<head><title> The story of Monkey </title></head>
```

用户可以用 BeautifulSoup 对象 soup 加标签名轻松地获取这些标签的内容，但应注意，它查找的是所有内容中第 1 个符合要求的标签，对于查询所有的标签，将在后面进行介绍。

可以验证一下这些对象的类型。

```
print(type(soup.title))     #输出：<class 'bs4.element.Tag'>
```

对于 Tag，它有两个重要的属性——name 和 attrs，下面分别来感受一下。

```
print(soup.name)              #输出：[document]
print(soup.head.name)         #输出：head
```

soup 对象本身比较特殊，它的 name 即为[document]，对于其他内部标签，输出的值便为标签本身的名称。

```
print(soup.p.attrs)           #输出：{'id': 'firstpara', 'align': 'center'}
```

在这里把 p 标签的所有属性打印输出，得到的类型是一个字典。

如果想要单独获取某个属性，例如获取它的 ID 可以这样做：

```
print(soup.p['id'])           #输出：firstpara
```

另外还可以利用 get()方法传入属性的名称，二者是等价的。

```
print(soup.p.get('id'))       #输出：firstpara
```

用户可以对这些属性和内容等进行修改，例如：

```
soup.p['class']="newClass"
```

另外还可以对这个属性进行删除，例如：

```
del soup.p['class']
```

2）NavigableString 对象

既然已经得到了标签的内容，要想获取标签内部的文字怎么办呢？很简单，用.string 即可，例如：

```
soup.title.string
```

这样就轻松获取到了标签里面的内容，如果用正则表达式则麻烦得多。

3）BeautifulSoup 对象

BeautifulSoup 对象表示的是一个文档的全部内容。大部分时候可以把它当作 Tag 对象，它是一个特殊的 Tag，下面的代码可以分别获取它的类型、名称以及属性。

```
print(type(soup))             #输出：<class 'bs4.BeautifulSoup'>
print(soup.name)              #输出：[document]
print(soup.attrs)             #输出空字典：{}
```

4）Comment 对象

Comment（注释）对象是一个特殊类型的 NavigableString 对象，其内容不包括注释符号，如果不好好地处理它，可能会对文本处理造成意想不到的麻烦。

6.3.4 用 BeautifulSoup 库操作解析 HTML 文档树

❶ 遍历文档树

1）用.content 属性和.children 属性获取直接子结点

Tag 的.content 属性可以将 Tag 的子结点以列表的方式输出。

```
print(soup.body.contents)
```

输出：

```
[<p align="center" id="firstpara">This is one paragraph</p>,
 <p align="center" id="secondpara">This is two paragraph</p>]
```

此时输出为列表，可以用列表索引来获取它的某一个元素。

```
print(soup.body.contents[0])     #获取第 1 个<p>
```

输出：

```
<p align="center" id="firstpara">This is one paragraph </p>
```

.children 属性返回的不是一个 list，它是一个 list 生成器对象，不过用户可以通过遍历获取所有子结点。

```
for child in soup.body.children:
    print(child)
```

输出：

```
<p align="center" id="firstpara"> This is one paragraph </p>
<p align="center" id="secondpara">This is two paragraph </p>
```

2）用.descendants 属性获取所有子孙结点

.contents 和.children 属性仅包含 Tag 的直接子结点，.descendants 属性可以对所有 Tag 的子孙结点进行递归循环，和.children 类似，用户也需要遍历获取其中的内容。

```
for child in soup.descendants:
   print(child)
```

可以发现，所有的结点都被打印出来，先是最外层的 HTML 标签，其次从 head 标签一个个剥离，依此类推。

3）结点内容

如果一个标签里面没有标签了，那么.string 就会返回标签里面的内容。如果标签里面只有唯一的一个标签，那么.string 也会返回最里面标签的内容。

如果 Tag 包含了多个子标签结点，Tag 将无法确定.string 方法应该调用哪个子标签结点的内容，.string 的输出结果是 None。

```
print(soup. title.string)       #输出<title>标签里面的内容
print(soup. body.string)        #<body>标签包含了多个子结点，所以输出 None
```

输出：

```
The story of Monkey
None
```

4）父结点

.parent 属性用于获取父结点。

```
p=soup.title
print(p.parent.name)        #输出父结点名 Head
```

输出：

```
Head
```

以上是遍历文档树的基本用法。

❷ 搜索文档树

1）find_all(name, attrs, recursive, text, **kwargs)

find_all()方法搜索当前 Tag 的所有 Tag 子结点，并判断是否符合过滤器的条件，其参数如下。

（1）name 参数：可以查找所有名字为 name 的标签。

```
print(soup.find_all('p'))     #输出所有<p>标签
[<p align="center" id="firstpara">This is one paragraph</p>, <p align=
"center" id="secondpara">This is two paragraph</p>]
```

如果 name 参数传入正则表达式作为参数，BeautifulSoup 会通过正则表达式的 match()来匹配内容。下面的例子找出所有以 h 开头的标签。

```
for tag in soup.find all(re.compile("^h")):
    print(tag.name, end=" ")       #html head
```

输出：

```
html head
```

这表示< html >和< head >标签都被找到。

（2）attrs 参数：按照 tag 标签属性值检索，需要列出属性名和值，采用字典形式。

```
soup.find all('p',attrs={'id':"firstpara"})或者 soup.find all('p', {'id':
"firstpara"})
```

它们都是查找属性值 id 是"firstpara"的<p>标签。

当然也可以采用关键字形式“soup.find_all('p', {id="firstpara"})”。

（3）recursive 参数：在调用 Tag 的 find_all()方法时 BeautifulSoup 会检索当前 Tag 的所有子孙结点，如果只想搜索 Tag 的直接子结点，可以使用 recursive=False。

（4）text 参数：通过 text 参数可以搜索文档中的字符串内容。

```
print(soup.find_all(text=re.compile("paragraph")))#re.compile()正则表达式
```

输出：

```
['This is one paragraph', 'This is two paragraph']
```

re.compile("paragraph")为正则表达式，表示所有含有“paragraph”的字符串都匹配。

（5）limit 参数：find_all()方法返回全部的搜索结构，如果文档树很大，那么搜索会很慢。如果用户不需要全部结果，可以使用 limit 参数限制返回结果的数量，当搜索到的结果数量达到 limit 的限制时就停止搜索返回结果。

文档树中有两个 Tag 符合搜索条件，但结果只返回了 1 个，因为限制了返回数量。

```
soup.find_all("p", limit=1)
```

输出：

```
[<p align="center" id="firstpara">This is one paragraph</p>]
```

2）find(name, attrs, recursive, text)

它与 find_all()方法唯一的区别是 find_all()方法返回全部结果的列表，而 find()方法返回找到的第 1 个结果。

❸ 用 CSS 选择器筛选元素

在写 CSS 时标签名不加任何修饰，类名前加点，ID 名前加#。在这里也可以利用类似的方法来筛选元素，用到的方法是 soup.select()，返回类型是列表。

（1）通过标签名查找：

```
soup.select('title')              #选取<title>元素
```

（2）通过类名查找：

```
soup.select('.firstpara')         #选取 class 是 firstpara 的元素
soup.select_one(".firstpara")     #查找 class 是 firstpara 的第 1 个元素
```

（3）通过 id 名查找：

```
soup.select('#firstpara')         #选取 id 是 firstpara 的元素
```

以上的 select()方法返回的结果都是列表形式，可以用遍历形式输出，然后用 get_text()方法或 text 属性来获取它的内容。

```
soup=BeautifulSoup(html, 'html.parser')
print type(soup.select('div'))
print(soup.select('div')[0].get_text())          #输出首个<div>元素的内容
for title in soup.select('title'):
  print(title.text)                              #输出所有<div>元素的内容
```

处理网页需要对 HTML 有一定的了解，BeautifulSoup 库是一个非常完备的 HTML 解析函数库，有了 BeautifulSoup 库的知识，就可以进行网络爬虫实战了。

```
from bs4 import BeautifulSoup
def getHtmlCode(url):             #该方法传入 url，返回 url 的 html 的源代码
    headers={
     'User-Agent': 'MMozilla/5.0(Windows NT 6.1; WOW64; rv:31.0) Gecko/
     20100101 Firefox/31.0'
    }
    url1=urllib.request.Request(url, headers=headers)
                                  #Request()函数将 url 添加到头部，模拟浏览器访问
    page=urllib.request.urlopen(url1).read()
                                  #将 url 页面的源代码保存成字符串
    page=page.decode('UTF-8')     #字符串转码
```

```
    return page

def getImg(page,localPath):
                #该方法传入 html 的源代码，截取其中的 img 标签，将图片保存到本机
    soup=BeautifulSoup(page,'html.parser')  #按照 html 格式解析页面
    imgList=soup.find all('img')            #返回包含所有 img 标签的列表
    x=0
    for imgUrl in imgList:                  #列表循环
        print('正在下载：%s'%imgUrl.get('src'))
        #urlretrieve(url,local)方法根据图片的 url 将图片保存到本机
        urllib.request.urlretrieve(imgUrl.get('src'),localPath+'%d.jpg'%x)
        x+=1
if __name__=='__main__':
    url='http://www.zhangzishi.cc/20160928gx.html'
    localPath='E:/img/'
    page=getHtmlCode(url)
    getImg(page,localPath)
```

可见使用 BeautifulSoup 能比使用正则表达式更简单地找到所有 img 标签。

6.3.5 requests 库的使用

requests 库和 urllib 库的作用相似且使用方法基本一致，都是根据 HTTP 协议操作各种消息和页面，但使用 requests 库比使用 urllib 库更简单些。

❶ requests 库的安装

使用 pip 直接安装 requests：

```
pip3 install requests
```

安装后进入 Python 导入模块测试是否安装成功。

```
import requests
```

没有出错即安装成功。

对于 requests 库的使用，请读者参阅“http://cn.python-requests.org/zh_CN/latest/”。

❷ 发送请求

发送请求很简单，首先要导入 requests 模块：

```
>>>import requests
```

接下来获取一个网页，例如中原工学院的首页：

```
>>>r=requests.get('http://www.zut.edu.cn')
```

之后就可以使用这个 r 的各种方法和函数了。

另外，HTTP 请求还有很多类型，例如 POST、PUT、DELETE、HEAD、OPTIONS，

可以用同样的方式实现：

```
>>> r=requests.post("http://httpbin.org/post")
>>> r=requests.head("http://httpbin.org/get")
```

❸ 在 URL 中传递参数

有时候需要在 URL 中传递参数，比如在采集百度搜索结果时，对于 wd 参数（搜索词）和 rn 参数（搜索结果数量），可以通过字符串连接的形式手工组成 URL，但 requests 提供了一种简单的方法：

```
>>> payload={'wd': '夏敏捷', 'rn': '100'}
>>> r=requests.get("http://www.baidu.com/s", params=payload)
>>> print(r.url)
```

结果如下：

```
http://www.baidu.com/s?wd=%E5%A4%8F%E6%95%8F%E6%8D%B7&rn=100
```

上面 wd=的乱码就是“夏敏捷”的 URL 转码形式。

POST 参数请求例子如下：

```
requests.post('http://www.itwhy.org/wp-comments-post.php', data=
{'comment': '测试 POST'})          #POST 参数
```

❹ 获取响应内容

```
>>> r=requests.get('http://www.baidu.com')     #返回一个 Response 对象 r
>>> r.text
```

在使用 requests()方法后会返回一个 Response 对象，其存储了服务器响应的内容，如上实例中已经提到的 r.text。

用户可以通过 r.text 来获取网页的内容。

结果如下：

```
'<!DOCTYPE html>\r\n<!--STATUS OK--><html> <head><meta http-equiv=content-
type content=text/html;charset=utf-8><meta http-equiv=X-UA-Compatible
content=IE=Edge><meta content=always name=referrer>...'
```

另外，还可以通过 r.content 来获取页面内容。

```
>>> r.content
```

r.content 以字节的方式去显示，所以在 IDLE 中以 b 开头。

```
>>> r.encoding      #可以使用 r.encoding 来获取网页编码
```

结果如下：

```
'utf-8'
```

当发送请求时，requests 会根据 HTTP 头部来获取网页编码；当使用 r.text 时，requests 就会使用这个编码。当然，用户还可以修改 requests 的编码形式。

```
>>> r=requests.get('http://www.zhidaow.com')
>>> r.encoding
'utf-8'
>>>r.encoding='ISO-8859-1'
```

像上面的例子，对 encoding 修改后直接用修改后的编码去获取网页内容。

❺ **JSON**

如果用到 JSON，就要引入新模块，例如 json 和 simplejson，但在 requests 中已经有了内置的函数 r.json()。这里以查询 IP 的 API 为例：

```
>>>r=requests.get('http://ip.taobao.com/service/getIpInfo.php?ip=
202.196.32.7')
>>> r.json()
{'data': {'region id': '410000', 'county': 'XX', 'city id': '410100',
'area': '', 'country': '中国', 'country_id': 'CN', 'isp': '教育网', 'area_id':
'', 'city': '郑州', 'ip': '202.196.32.7', 'region': '河南', 'isp_id':
'100027', 'county id': 'xx'}, 'code': 0}
>>>r.json()['data']['country']
'中国'
```

❻ **网页状态码**

用户可以使用 r.status_code 来检查网页的状态码。

```
>>>r=requests.get('http://www.mengtiankong.com')
>>>r.status code
200
>>>r=requests.get('http://www.mengtiankong.com/123123/')
>>>r.status code
404
```

此时，能正常打开网页的返回 200，不能正常打开的返回 404。

❼ **响应的头部内容**

用户可以通过 r.headers 来获取响应的头部内容。

```
>>>r=requests.get('http://www.zhidaow.com')
>>> r.headers
{
    'content-encoding': 'gzip',
    'transfer-encoding': 'chunked',
    'content-type': 'text/html; charset=utf-8';
    …
}
```

可以看到以字典的形式返回了全部内容，用户也可以访问部分内容。

```
>>> r.headers['Content-Type']
'text/html; charset=utf-8'
>>> r.headers.get('content-type')
```

```
'text/html; charset=utf-8'
```

❽ 设置超时时间

用户可以通过 timeout 属性设置超时时间，一旦超过这个时间还没有获得响应内容，就会提示错误。

```
>>> requests.get('http://github.com', timeout=0.001)
Traceback(most recent call last):
  File "<stdin>", line 1, in <module>
requests.exceptions.Timeout: HTTPConnectionPool(host='github.com',
port=80): Request timed out.(timeout=0.001)
```

❾ 代理访问

在采集时为避免被封 IP，经常会使用代理。requests 也有相应的 proxies 属性。

```
import requests
proxies={
  "http": "http://10.10.1.10:3128",
  "https": "http://10.10.1.10:1080",
}
requests.get("http://www.zhidaow.com", proxies=proxies)
```

如果代理需要账户和密码，则需要这样：

```
proxies={
    "http": "http://user:pass@10.10.1.10:3128/",
}
```

❿ 请求头内容

请求头内容可以用 r.request.headers 来获取。

```
>>>r.request.headers
{'Accept-Encoding': 'identity, deflate, compress, gzip',
'Accept': '*/*', 'User-Agent': 'python-requests/1.2.3 CPython/2.7.3 Windows/XP'}
```

⓫ 自定义请求头部

伪装请求头部是爬虫采集信息时经常用到的，用户可以用这个方法来隐藏自己：

```
>>>r=requests.get('http://www.zhidaow.com')
>>>print(r.request.headers['User-Agent'])  #输出 python-requests/2.13.0
>>>headers={'User-Agent': 'xmj'}
>>>r=requests.get('http://www.zhidaow.com', headers=headers)
                                           #伪装的请求头部
>>>print(r.request.headers['User-Agent'] ) #输出 xmj，避免被反爬虫
```

再例如另一个定制 header 的例子：

```
import  requests
import  json
data={'some': 'data'}
```

```
headers={'content-type': 'application/json',
        'User-Agent': 'Mozilla/5.0(X11; Ubuntu; Linux x86 64; rv:22.0)
        Gecko/20100101 Firefox/22.0'}
r=requests.post('https://api.github.com/some/endpoint', data=data,
headers=headers)
print(r.text)
```

下面用 requests 库替换 urllib 库，并用 open().write()方法替换掉 urllib.request.urlretrieve (url, localPath)方法来下载中原工学院主页上的所有图片。

```
'''''
    使用 requests、bs4 库下载中原工学院主页上的所有图片
'''
import os
import requests
from bs4 import BeautifulSoup
def getHtmlCode(url):         #该方法传入 url，返回 url 的 html 的源代码
    headers={
     'User-Agent': 'MMozilla/5.0(Windows NT 6.1; WOW64; rv:31.0) Gecko/
     20100101 Firefox/31.0'
    }
    r=requests.get(url,headers=headers)
    r.encoding='UTF-8'                     #指定网页解析的编码格式
    page=r.text                            #获取 url 页面的源代码字符串文本
    return page

def getImg(page,localPath):  #该方法传入 html 的源代码，截取其中的 img 标签，将图
                             #片保存到本机
    if not os.path.exists(localPath):  #新建文件夹
        os.mkdir(localPath)
    soup=BeautifulSoup(page,'html.parser') #按照 html 格式解析页面
    imgList=soup.find_all('img')     #返回包含所有 img 标签的列表
    x=0
    for imgUrl in imgList:                 #列表循环
        try:
            print('正在下载：%s'%imgUrl.get('src'))
            if "http://" not in imgUrl.get('src'):  #不是绝对路径 http 开始
                m='http://www.zut.edu.cn/'+imgUrl.get('src')
                print('正在下载：%s'%m)
                ir=requests.get('http://www.zut.edu.cn/'+imgUrl.get('src'))
            else:
                ir=requests.get(imgUrl.get('src'))
            #用 write()方法写入本地文件中
            open(localPath+'%d.jpg'%x, 'wb').write(ir.content)
            x+=1
        except:
```

```
            continue
if __name__=='__main__':
    url=' http://www.zut.edu.cn/'
    localPath='E:/img/'
    page=getHtmlCode(url)
    getImg(page,localPath)
```

掌握上述技术后先爬取较简单的搜狗图片中某主题的图片。

输入搜狗图片的网址“http://pic.sogou.com/”，进入壁纸分类，然后按 F12 键进入开发人员选项（编者用的是 Google Chrome 浏览器）。右击某张图片，在快捷菜单中选择“检查”命令，结果如图 6-2 所示。

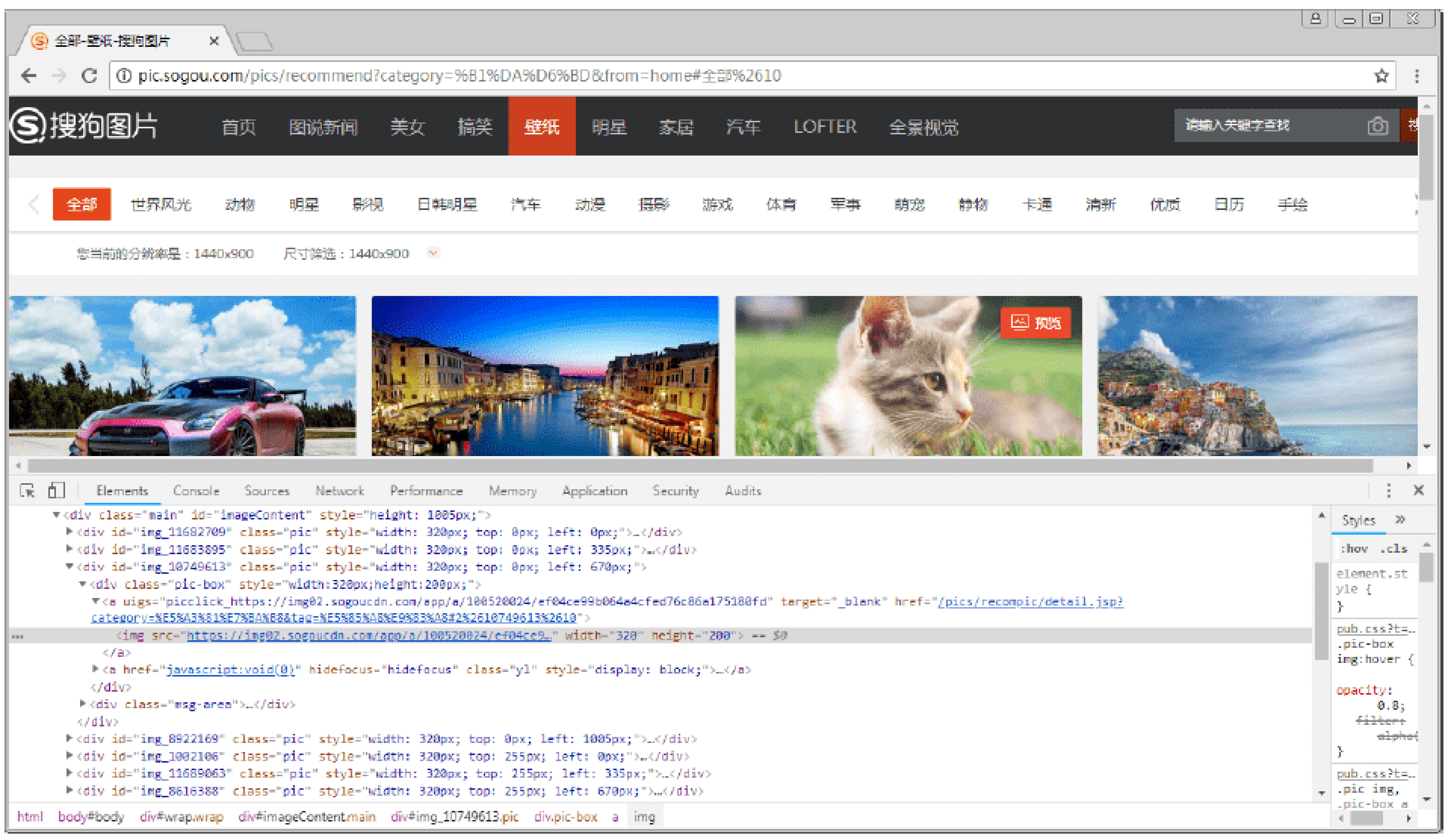


图 6-2　网页代码示意图

发现需要的图片是在 img 标签下的，于是先试着用 Python 的 requests 提取该组件，进而获取 img 的 src，然后使用 urllib.request.urlretrieve 逐个下载图片，从而达到批量获取资料的目的。爬取的 URL 如下：

http://pic.sogou.com/pics/recommend?category=%B1%DA%D6%BD

此 URL 来自进入分类后的浏览器的地址栏。

写出如下代码：

```
import requests
import urllib
from bs4 import BeautifulSoup
res=requests.get('http://pic.sogou.com/pics/recommend?category=
%B1%DA%D6%BD')        #爬取的 URL
soup=BeautifulSoup(res.text,'html.parser')
print(soup.select('img'))
```

输出：

```
[<img alt="搜狗图片" src="/news/images/tupian130x34_@1x.png" srcset=
"/news/images/tupian130x34_@2x.png 2x"/>, <img class="st-load" src=""/>]
```

此时发现输出内容并不包含想要的图片元素，而是只剖析到 Logo（见图 6-3）的 img，这显然不是大家想要的。也就是说，需要的图片资料不在“http://pic.sogou.com/pics/recommend?category= %B1%DA%D6%BD”的 HTML 源代码里面。



图 6-3　tupian130x34_@1x.png

这是为什么呢？可以发现当在网页内向下滑动鼠标滚轮时图片是动态刷新出来的，也就是说，该网页并不是一次加载出全部资源，而是动态加载资源。这也避免了因为网页过于臃肿而影响加载速度。在网页动态加载中找出图片元素的方法如下：

按 F12 键，在 Network 的 XHR 下单击文件链接，在 Preview 选项卡中观察结果，如图 6-4 所示。

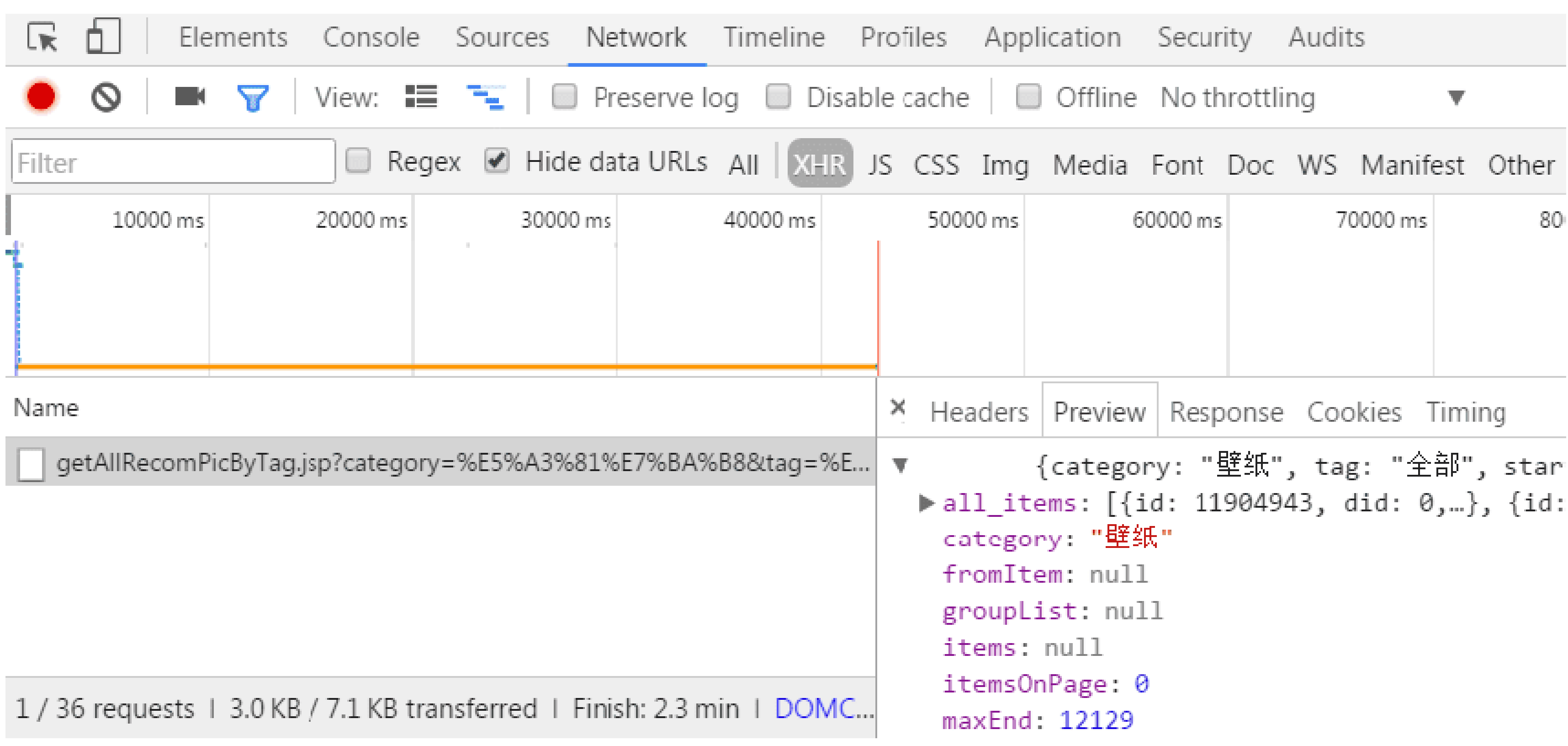


图 6-4　分析网页的 JSON 数据

说明：XHR 的全称为 XMLHttpRequest，中文解释为可扩展超文本传输请求。其中，XML 是可扩展标记语言，Http 是超文本传输协议，Request 是请求。XMLHttpRequest 对象可以在不向服务器提交整个页面的情况下实现局部更新网页。当页面全部加载完毕后，客户端通过该对象向服务器请求数据，服务器端接收数据并处理后向客户端反馈数据。XMLHttpRequest 对象提供了对 HTTP 协议的完全访问，包括做出 POST 和 HEAD 请求以及普通的 GET 请求的能力。XMLHttpRequest 可以同步或异步返回 Web 服务器的响应，并且能以文本或者一个 DOM 文档的形式返回内容。尽管名为 XMLHttpRequest，但它并不限于和 XML 文档一起使用，它可以接收任何形式的文本文档。XMLHttpRequest 对象是为 AJAX 的 Web 应用程序架构的一项关键功能。

单击图 6-4 中的 JSON 数据 all_items，发现下面是一个个貌似图片的元素。试着打开一个 URL，发现确实是图片的地址。找到目标之后单击 XHR 下的 Headers 得到：

http://pic.sogou.com/pics/channel/getAllRecomPicByTag.jsp?category=%E5%A3%81%E7%BA%B8&tag=%E5%85%A8%E9%83%A8&start=0&len=15&width=1536&height=864

试着去掉一些不必要的部分，技巧就是删掉可能的部分之后访问不受影响，最后得到

的 URL 如下：

http://pic.sogou.com/pics/channel/getAllRecomPicByTag.jsp?category=%E5%A3%81%E7%BA%B8&tag=%E5%85%A8%E9%83%A8&start=0&len=15

从字面意思知道 category 后面可能为分类。start 为开始下标，len 为长度，即图片的数量。通过这个 URL 请求得到响应的 JSON 数据中包含着用户所需要的图片地址。有了上面的分析可以写出以下代码：

```
import requests
import json
import urllib
def getSogouImag(category,length,path):
    n=length
    cate=category
    imgs=requests.get('http://pic.sogou.com/pics/channel/
    getAllRecomPicByTag.jsp?category='+cate+'&tag=%E5%85%A8%E9%83%A8&start=
    0&len='+str(n))
    jd=json.loads(imgs.text)
    jd=jd['all items']
    imgs url=[]
    for j in jd:
        imgs url.append(j['bthumbUrl'])
    m=0
    for img url in imgs url:
            print('***** '+str(m)+'.jpg *****'+'Downloading…')
            urllib.request.urlretrieve(img url,path+str(m)+'.jpg')
            m=m+1
    print('Download complete!')
getSogouImag('壁纸',200,'D:/download/壁纸/')
                                    #下载 200 张图片到 D 盘 download 下的“壁纸”文件夹中
```

程序运行结果如图 6-5 所示。

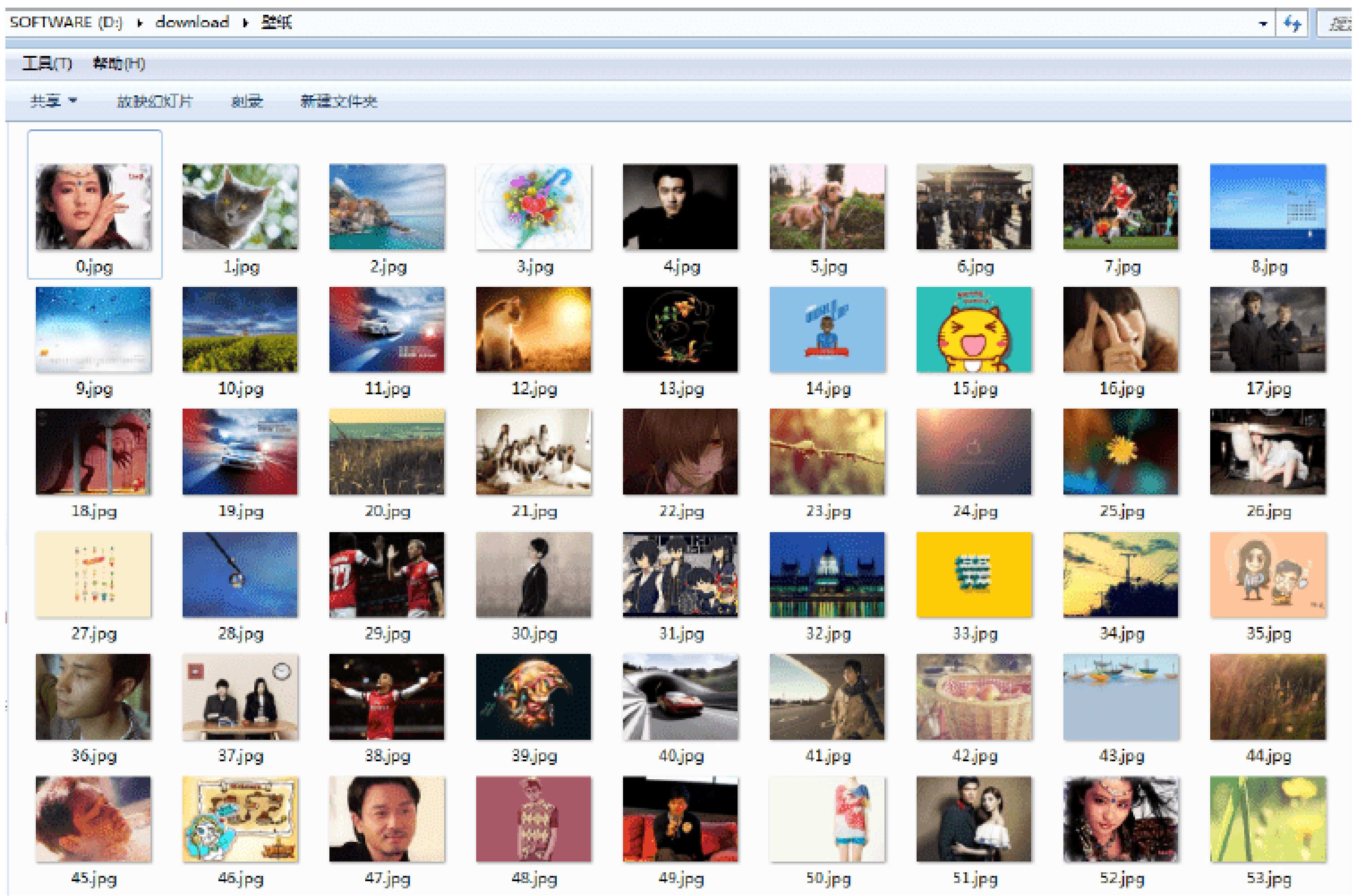


图 6-5　爬取到 D 盘 download 下“壁纸”文件夹中的图片

至此，关于该爬虫程序的编程介绍完毕。从整体来看，找到需要爬取元素所在的 url 是爬虫诸多环节中的关键。

有了用搜狗图片下载图片的基础，下面来实现百度图片的图片下载。

6.4　程序设计的步骤

6.4.1　分析网页源代码和网页结构

进入百度图片界面（https://image.baidu.com/），输入某个关键字（例如夏敏捷），然后单击“百度一下”按钮搜索，可见如下网址：

https://image.baidu.com/search/index?tn=baiduimage&ct=201326592&lm=-1&cl=2&ie=gbk&word=%CF%C4%C3%F4%BD%DD&fr=ala&ala=1&alatpl=adress&pos=0&hs=2&xthttps=111111

其中，%CF%C4%C3%F4%BD%DD 就是“夏敏捷”的 URL 编码（网址上不使用汉字），所看见的页面是“瀑布流版本”（如图 6-6 所示），当向下滑动的时候可以不停刷新，这是一个动态的网页（和搜狗图片类似，需要按 F12 键，通过 Network 下的 XHR 去分析网页的结构），而用户可以选择更简单的方法，就是单击网页右上方的“传统翻页版本”（如图 6-7 所示）。



图 6-6　瀑布流版本下的图片

在传统翻页版本下的浏览器地址栏中可见如下网址：

https://image.baidu.com/search/flip?tn=baiduimage&ie=gbk&word=%CF%C4%C3%F4%BD%DD&ct=201326592&lm=-1&v=flip

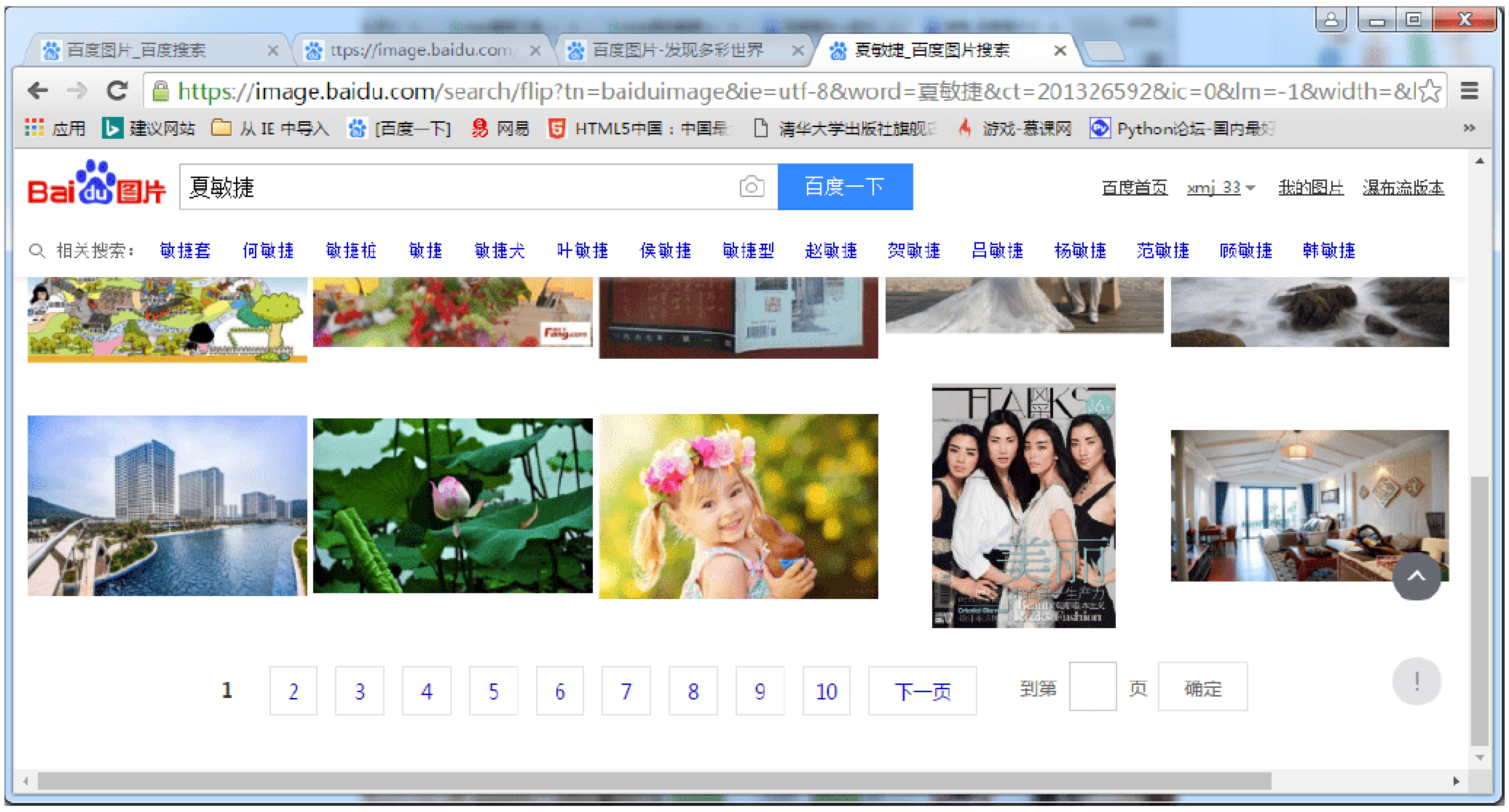


图 6-7　传统翻页版本下的图片

在传统翻页版本下单击“下一页”或某数字页码，网址会发生变化，而动态网页则不会，因为其分页参数是在 POST 请求中的。在该程序中使用这个网址请求页面。

在网页空白处右击选择“查看网页的源代码”命令，可以查看网页的源代码（如图 6-8 所示），也就是 requests get 下来的数据，在这里面找到各个图片的链接和下一页的链接比较困难。

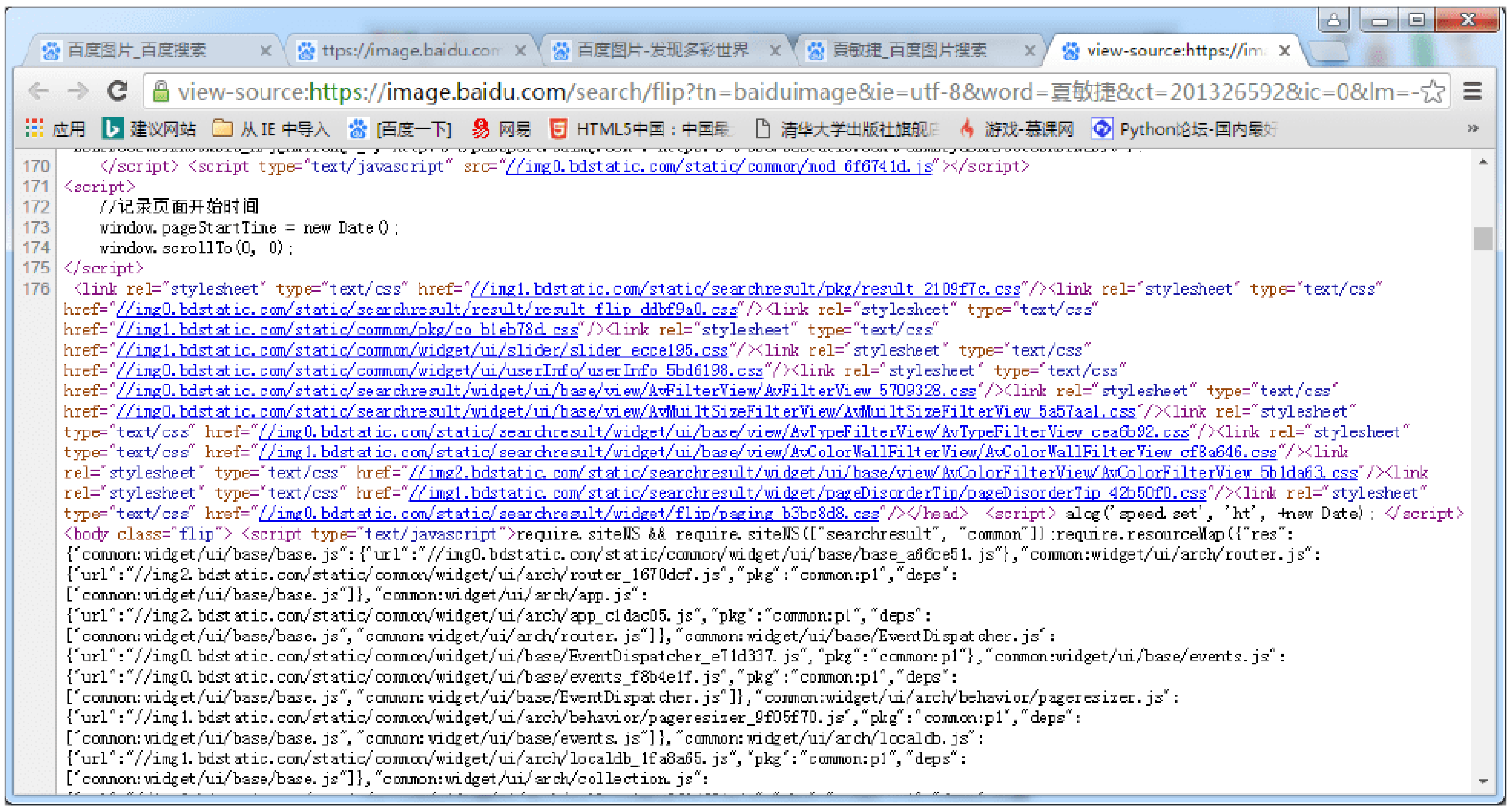


图 6-8　传统翻页版本下的图片

用户可以通过浏览器（例如 Chrome）的开发者工具来查看网页的元素，按 F12 键打开开发者工具来查看网页样式，注意当鼠标从结构表中滑过时会实时显示此段代码所对应的位置区域（注意先要单击开发者工具右上角的箭头按钮），用户可以通过此方法快速地找到图片元素所对应的位置（如图 6-9 所示）。

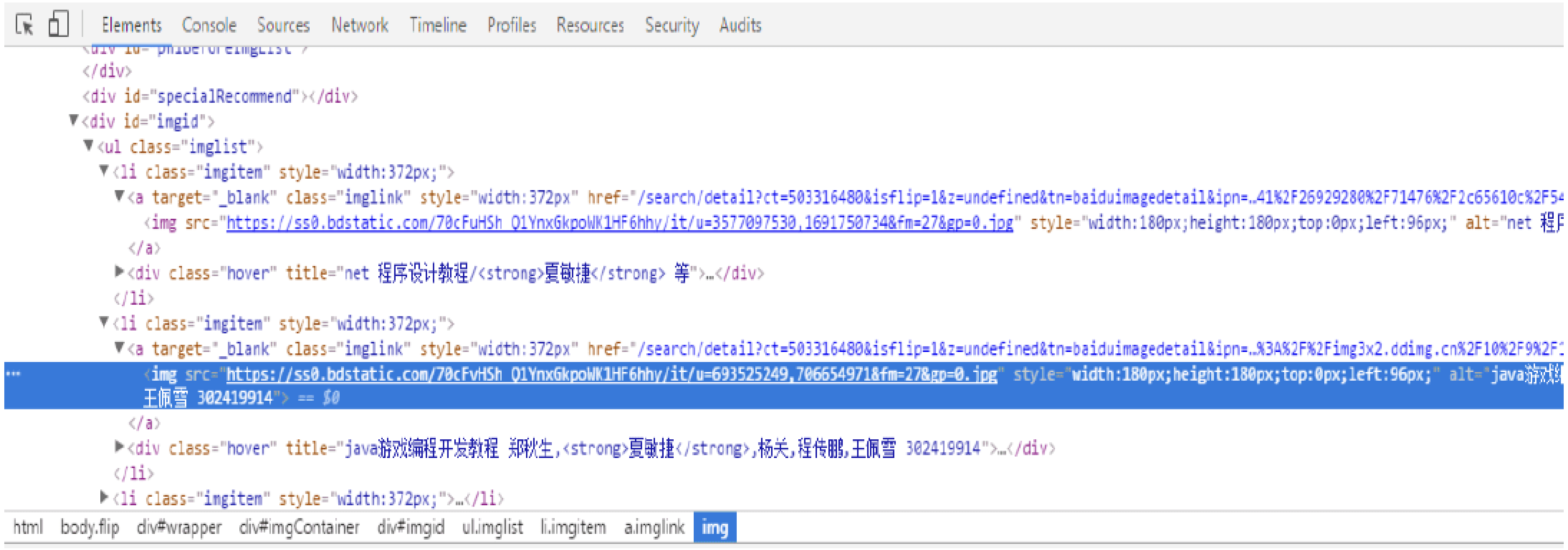


图 6-9　图片元素所对应的位置

对图 6-9 分析可知，每个图片都在<ul class="imglist">下的列表项<li class="imgitem" style="width:372px;">中，其中<img src="…">保存图片的网址。

```
<div id="imgid">
  <ul class="imglist">
    <li class="imgitem" style="width:372px;">
      <a target=" blank"
        <img src="https://ss0.bdstatic.com/70cFuHSh Q1YnxGkpoWK1HF6hhy/it/
        u=3577097530,1691750734&amp;fm=27&amp;gp=0.jpg" alt="net 程序设计教程">
      </a>
    <div class="hover" title="net 程序设计教程/<strong>夏敏捷</strong>
    等"></div>
</li>
```

从上面找到了一张图片的路径：

https://ss0.bdstatic.com/70cFuHSh_Q1YnxGkpoWK1HF6hhy/it/u=3577097530,1691750734&fm=27&gp=0.jpg

用户可以在 HTML 源代码中搜索此路径找到它的位置，如下：

```
flip.setData('imgData',
{ "queryEnc":"%E5%A4%8F%E6%95%8F%E6%8D%B7", "displayNum":5722, "bdIsClustered" :
"1", "listNum":1977, "bdFmtDispNum" : "5722", "bdSearchTime" : "",
"isNeedAsyncRequest":0,
"data":[{"thumbURL":"https://ss0.bdstatic.com/70cFuHSh
Q1YnxGkpoWK1HF6hhy/it/u=3577097530,1691750734&fm=27&gp=0.jpg",
"middleURL":"https://ss0.bdstatic.com/70cFuHSh Q1YnxGkpoWK1HF6hhy/it/u=
3577097530,1691750734&fm=27&gp=0.jpg", "largeTnImageUrl":"", "hasLarge" :0,
"hoverURL":"https://ss0.bdstatic.com/70cFuHSh Q1YnxGkpoWK1HF6hhy/it/u=
3577097530,1691750734&fm=27&gp=0.jpg", "pageNum":0,
"objURL":"http://img13.360buyimg.com/n0/jfs/t586/241/26929280/71476/
2c65610c/54484fe6Nb33010bd.jpg",
"fromURL":"ippr z2C$qAzdH3FAzdH3Ftpj4 z&e3B31 z&e3Bv54AzdH3F8nc9adan0n
z&e3Bip4s", "fromURLHost":"item.jd.com", "currentIndex":"", "width":800,
```

```
"height":800, "type":"jpg", "filesize":"", "bdSrcType":"0",
"di":"35266154990", "pi":"0", "is":"0,0", "partnerId":0, "bdSetImgNum":0,
"bdImgnewsDate":"1970-01-01 08:00",
```

可见“thumbURL”“middleURL”和“objURL”均是图片的所在网址，这里选用“objURL”对应的网址图片，所以写出如下正则表达式获取图片的所在网址：

```
re.findall('"objURL":"(.*?)"',content,re.S)
```

通过分析可知，“下一页”或某数字页码 HTML 的代码如下：

```
<div id="page">
<strong><span class="pc">1</span></strong>
<a href="/search/flip?tn=baiduimage&ie=utf-8&word=%E5%A4%8F%E6%95%8F%E6%8D%B7
&pn=20&gsm=3c&ct=&ic=0&lm=-1&width=0&height=0"><span class="pc" data="right">
2</span></a>
<a href="/search/flip?tn=baiduimage&ie=utf-8&word=%E5%A4%8F%E6%95%8F%E6%8D%B7
&pn=40&gsm=0&ct=&ic=0&lm=-1&width=0&height=0"><span class="pc" data=
"right">3</span></a>
…
<a href="/search/flip?tn=baiduimage&ie=utf-8&word=%E5%A4%8F%E6%95%8F%E6%8D%B7
&pn=180&gsm=0&ct=&ic=0&lm=-1&width=0&height=0"><span class="pc" data=
"right">10</span></a>
<a href="/search/flip?tn=baiduimage&ie=utf-8&word=%E5%A4%8F%E6%95%8F%E6%8D%B7
&pn=20&gsm=3c&ct=&ic=0&lm=-1&width=0&height=0" class="n">下一页</a>
</div>
```

所以获取“下一页”链接写出如下正则表达式：

```
re.findall('<div id="page">.*<a href="(.*?)" class="n">',content,re.S)[0]
```

6.4.2 设计代码

视频讲解

Python 爬虫搜索百度图片库并下载图片的代码如下：

```
import requests                  #首先导入库
import re
#设置默认配置
MaxSearchPage=20                 #搜索页数
CurrentPage=0                    #当前正在搜索的页数
DefaultPath="pictures"           #默认储存位置
NeedSave=0                       #是否需要储存
#图片链接正则和下一页的链接正则
def imageFiler(content):         #通过正则获取当前页面的图片地址数组
        return re.findall('"objURL":"(.*?)"',content,re.S)
def nextSource(content):         #通过正则获取下一页的网址
        next=re.findall('<div id="page">.*<a href="(.*?)" class="n">',
        content,re.S)[0]
```

```
        print("---------"+"http://image.baidu.com"+next)
        return next
#爬虫主体
def spidler(source):
        content=requests.get(source).text      #通过链接获取内容
        imageArr=imageFiler(content)       #获取图片数组
        global CurrentPage
        print("Current page:"+str(CurrentPage)+"**************")
        for imageUrl in imageArr:
            print(imageUrl)
            global  NeedSave
            if NeedSave:              #如果需要保存图片则下载图片，否则不下载图片
               global DefaultPath
               try:
                    #下载图片并设置超时时间，如果图片地址错误就不继续等待了
                    picture=requests.get(imageUrl,timeout=10)
               except:
                    print("Download image error! errorUrl:"+imageUrl)
                    continue
               #创建图片保存的路径
               imageUrl=imageUrl.replace('/','').replace(':','').
              replace('?','')
               pictureSavePath=DefaultPath+imageUrl
               fp=open(pictureSavePath,'wb')     #以写入二进制的方式打开文件
               fp.write(picture.content)
               fp.close()
        global MaxSearchPage
        if CurrentPage<=MaxSearchPage:         #继续下一页爬取
             if nextSource(content):
                 CurrentPage+=1
                 #爬取完毕后通过下一页地址继续爬取
                 spidler("http://image.baidu.com"+nextSource(content))
#爬虫的开启方法
def beginSearch(page=1,save=0,savePath="pictures/"):
        #(page:爬取页数,save:是否储存,savePath:默认储存路径)
        global MaxSearchPage,NeedSave,DefaultPath
        MaxSearchPage=page
        NeedSave=save                         #是否保存，值为 0 不保存，1 保存
        DefaultPath=savePath                  #图片保存的位置
        key=input("Please input you want search: ")
        StartSource="http://image.baidu.com/search/flip?tn=
        baiduimage&ie=utf-8&word="+str(key)+"&ct=201326592&v=flip"
       #分析链接可以得到，替换其"word"值后面的数据来搜索关键词
        spidler(StartSource)
#调用开启的方法就可以通过关键词搜索图片了
```

```
beginSearch(page=5,save=1)          #page=5 是下载前 5 页，save=1 为保存图片
```

运行后输入搜索关键词，例如“夏敏捷”，可以在 pictures 文件夹下得到夏敏捷的相关图片，如图 6-10 所示。这里下载的图片的命名采用的是下载的网址，所以需要去除文件名不允许的特殊字符，例如“:”“/”“?”等。当然，更好的处理方法是文件名采用数字编号，避免网址中出现特殊字符。

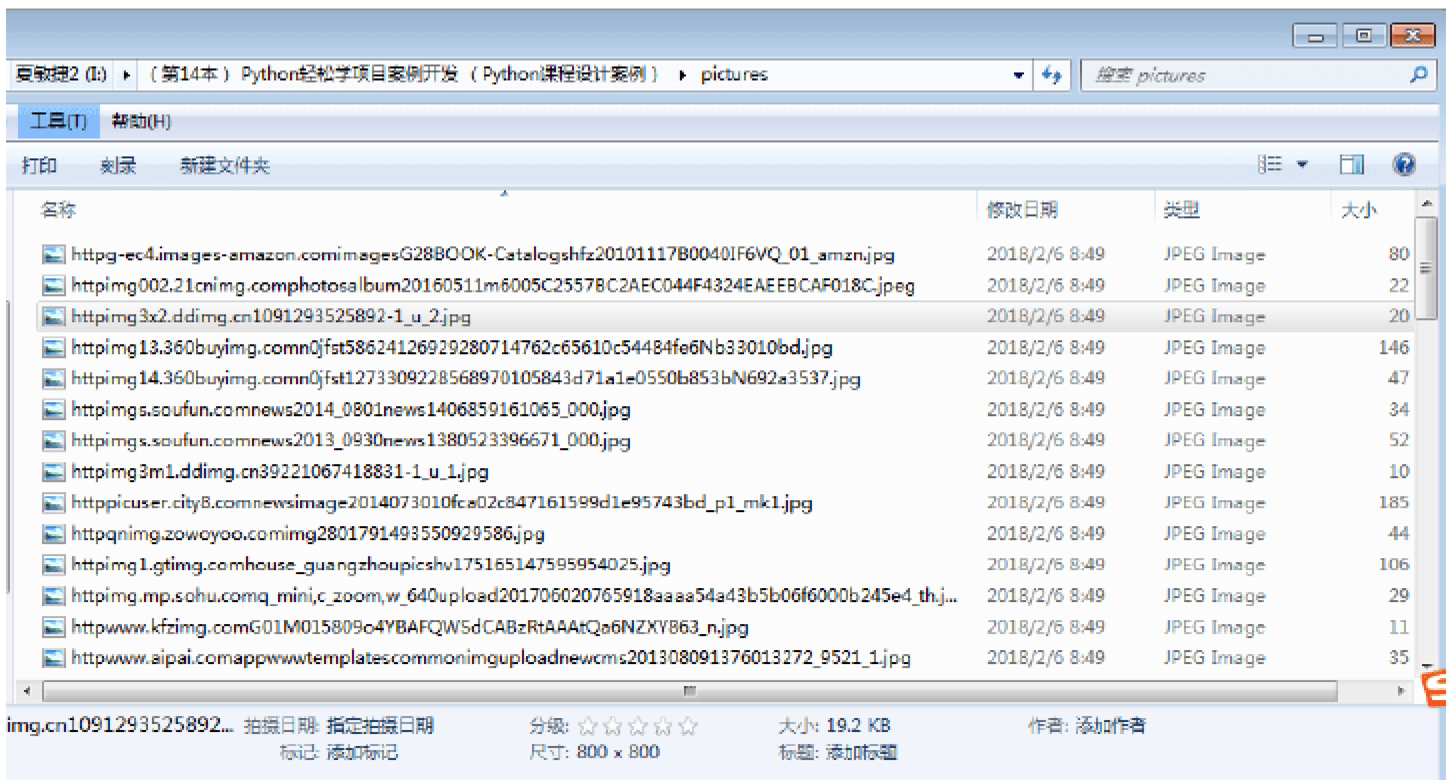


图 6-10　pictures 文件夹下得到相关图片

第 7 章

itchat 应用——微信机器人

7.1 itchat 功能介绍

视频讲解

本程序运行后会出现一张二维码图片，微信用户通过微信扫描二维码登录自己的微信，此时如果有好友发过来信息，则微信机器人会自动回复好友，效果如图 7-1 所示。在该图中可以看到当好友发过来“你好吗”，微信机器人会自动回复“还不错，你呢”；当好友发过来“讲个笑话吧”，微信机器人会自动回复一个笑话；当好友问“郑州天气？”时，微信机器人会自动回复天气情况，等等。

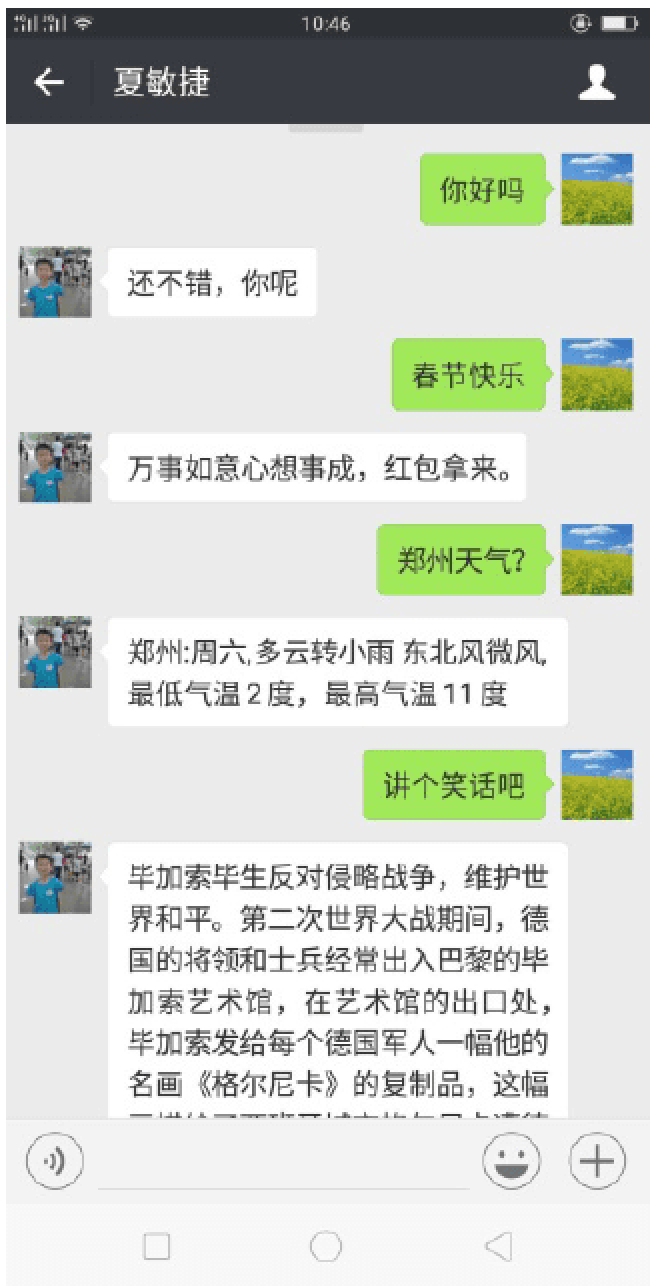


图 7-1　微信机器人聊天效果

7.2　程序设计的思路

该程序需要用到一个 Python 库——itchat，itchat 是一个开源的微信个人账号的接口，可以使用该库进行微信网页版中的所有操作；另外还需要用到图灵机器人 API，图灵机器人是一个中文语境下的对话机器人。本章主要使用 itchat 库和图灵机器人 API 完成一个能够处理微信消息的图灵机器人，包括好友聊天、群聊天。

7.3　关键技术

7.3.1　安装 itchat

itchat 是一个开源的微信个人账号的接口，使得 Python 调用微信功能从未如此简单，用户使用不到三十行的代码就可以完成一个能够处理所有信息的微信机器人。itchat 库已经做好了用代码调用微信的大多数功能，使用起来非常方便，官方技术文档的网址为“http://itchat.readthedocs.io/zh/latest/”，用户在使用时需要安装 itchat 库，在安装的时候使用 pip 即可。

```
pip install itchat
```

7.3.2　itchat 的登录微信

运行以下代码，会出现一张图 7-2 所示的二维码，扫码登录之后将会给“文件传输助手”发送一条“Hello, filehelper”的消息。

图 7-2　二维码

```
import itchat                                  #加载 itchat 库
itchat.auto_login()                            #登录微信
#发送文本消息，发送目标是“文件传输助手”
itchat.send('Hello, filehelper', toUserName='filehelper')
```

7.3.3 itchat 的消息类型

itchat 支持所有的消息类型与群聊。在 itchat 中定义了文本、图片、名片、位置、通知、分享、文件等多种消息类型，可以分别执行不同的处理。下面的示例注册了一个消息响应事件，用来定义接收到文本消息后如何处理。

```
import itchat
#注册消息响应事件，消息类型为itchat.content.TEXT，即文本消息，把装饰器写成下面的形
#式即可
@itchat.msg register(itchat.content.TEXT)
def text reply(msg):
  #返回同样的文本消息
  return msg['Text']

itchat.auto login()                    #登录微信
#绑定消息响应事件后让 itchat 运行起来，监听消息
itchat.run()
```

itchat.content 中包含所有的消息类型参数，如表 7-1 所示。

表 7-1 itchat.content 中所有的消息类型参数

参　数	类　型	Text 键值
TEXT	文本	文本内容（文字消息）
MAP	地图	位置文本（位置分享）
CARD	名片	推荐人字典（推荐人的名片）
SHARING	分享	分享名称（分享的音乐或者文章等）
PICTURE	图片/表情	下载方法
RECORDING	语音	下载方法
ATTACHMENT	附件	下载方法
VIDEO	小视频	下载方法
FRIENDS	好友邀请	添加好友所需参数
SYSTEM	系统消息	更新内容的用户或群聊的 UserName 组成的列表
NOTE	通知	通知文本（消息撤回等）

比如需要存储发送给自己的附件：

```
@itchat.msg register(ATTACHMENT)
def download files(msg):
    msg['Text'](msg['FileName'])
```

msg 字典的 Text 键是个下载方法（函数），可以下载文件。

再来看如何处理其他类型的消息，可以在消息响应事件里把 msg 打印出来，它是一个字典，看有哪些感兴趣的字段。下面演示了对这些消息类型的简单处理。

```
import itchat
from itchat.content import *              #import 全部消息类型
```

```
#处理文本类消息，包括文本、位置、名片、通知、分享
@itchat.msg_register([TEXT, MAP, CARD, NOTE, SHARING])
def text_reply(msg):
  #微信里的每个用户和群聊都使用 ID 来区分，msg['FromUserName']就是发送者的 ID
  #将消息的类型和文本内容返回给发送者
  itchat.send('%s: %s' % (msg['Type'], msg['Text']), msg['FromUserName'])

#处理多媒体类消息，包括图片、录音、文件、视频
@itchat.msg_register([PICTURE, RECORDING, ATTACHMENT, VIDEO])
def download_files(msg):
  #msg['Text']是一个文件下载函数，传入文件名，将文件下载下来
  msg['Text'](msg['FileName'])
  #把下载好的文件再发给发送者
  return '@%s@%s' % ({'Picture':'img','Video':'vid'}.get(msg['Type'],
  'fil'), msg['FileName'])

#处理好友添加请求，收到好友邀请自动添加好友
@itchat.msg_register(FRIENDS)
def add_friend(msg):
  itchat.add_friend(**msg['Text']) #该操作会自动将新好友的消息录入，不需要重载通讯录
  #加完好友后给好友打个招呼
  itchat.send_msg('Nice to meet you!', msg['RecommendInfo']['UserName'])

#处理群聊消息，在注册时增加 isGroupChat=True 将判定为群聊回复
@itchat.msg_register(TEXT, isGroupChat=True)
def text_reply(msg):
  if msg['isAt']:
    itchat.send(u'@%s\u2005I received: %s' % (msg['ActualNickName'],
    msg ['Content']), msg['FromUserName'])

#在 auto_login()里面提供一个 True，即 hotReload=True
#即可保留登录状态，即使程序关闭，在一定时间内重新开启也可以不重新扫码
itchat.auto_login(True)
itchat.run()
```

在 PICTURE、RECORDING、ATTACHMENT、VIDEO 几类的 msg 字典的 Text 键下存放了用于下载消息内容的函数，传入文件名即可下载，被发送的文件名都存储在 msg 的 FileName 键中。

区分群聊消息还是与好友聊天，在注册时增加 isGroupChat=True 将判定为群聊回复。

例如注册@itchat.msg_register(TEXT, isGroupChat=True)将判定为群聊回复，而注册@itchat.msg_register(TEXT)将判定为与好友聊天。

值得注意的是，群消息增加了 3 个键值：

```
isAt: 判断是否@本号自己；ActualNickName: 实际 NickName；Content:信息内容
```

可以通过以下程序测试：

```
import itchat
from itchat.content import TEXT
@itchat.msg register(TEXT, isGroupChat=True)
def text reply(msg):
    if(msg.isAt):        #判断是否有人@自己
        #如果有人@自己，就发一个消息告诉对方已经收到了信息
        itchat.send msg("我已经收到了来自{0}的消息，实际内容为{1}".format(msg
        ['ActualNickName'],msg['Text']),toUserName=msg['FromUserName'])
    print(msg.isAt)     #输出 True 或 False
    print(msg.actualNickName)
    print(msg.text)
itchat.auto login()
itchat.run()
```

7.3.4　itchat 回复消息

itchat 提供了 5 种回复方法，建议用户直接使用 send()方法。

❶ send()方法

格式：

```
send(msg='Text Message', toUserName=None)
```

参数：

- msg：发送消息的内容， '@fil@文件地址'将会被识别为传送文件，'@img@图片地址'将会被识别为传送图片，'@vid@视频地址'将会被识别为传送小视频。
- toUserName：发送对象，如果留空将会发送给自己。

返回值：

发送成功为 True，失败为 False。

程序示例：

```
import itchat
itchat.auto login()
itchat.send('Hello world!')
#请确保该程序目录下存在 gz.gif、xlsx.xlsx 和 demo.mp4 文件
itchat.send('@img@%s' % 'gz.gif')
itchat.send('@fil@%s' % 'xlsx.xlsx')
itchat.send('@vid@%s' % 'demo.mp4')
```

❷ send_msg()方法

格式：

```
send_msg(msg='Text Message', toUserName=None)
```

参数：

- msg：消息内容。
- toUserName：发送对象，如果留空将会发送给自己。

返回值：

发送成功为 True，失败为 False。

程序示例：

```
import itchat
itchat.auto_login()
itchat.send_msg('Hello world')
```

❸ send_file()方法

格式：

```
send_file(fileDir, toUserName=None)
```

参数：

- fileDir：文件路径（不存在该文件时将打印无此文件的提醒）。
- toUserName：发送对象，如果留空将会发送给自己。

返回值：

发送成功为 True，失败为 False。

程序示例：

```
import itchat
itchat.auto_login()
#请确保该程序目录下存在 xlsx.xlsx 文件
itchat.send_file('xlsx.xlsx')
```

❹ send_img()方法

格式：

```
send_img(fileDir, toUserName=None)
```

参数：

- fileDir：文件路径（不存在该文件时将打印无此文件的提醒）。
- toUserName：发送对象，如果留空将会发送给自己。

返回值：

发送成功为 True，失败为 False。

程序示例：

```
itchat.send_img('gz.gif')
```

❺ send_ video()方法

格式：

```
send_video(fileDir, toUserName=None)
```

- fileDir：文件路径（不存在该文件时将打印无此文件的提醒）。
- toUserName：发送对象，如果留空将会发送给自己。

返回值：

发送成功为 True，失败为 False。

程序示例（需要保证发送的视频为一个实际存在的 MP4 文件）：

```
itchat.send_file('demo.mp4')                  #请确保该程序目录下存在 demo.mp4 文件
```

7.3.5　itchat 获取账号

在使用个人微信的过程中主要有 3 种账号需要获取，分别为好友、公众号、群聊。itchat 为这 3 种账号提供了整体获取方法与搜索方法，而群聊多出获取用户列表的方法以及创建群聊、增加和删除用户的方法。这里分别介绍这 3 种账号的使用方法。

❶ 好友

好友的获取方法为 get_friends()，将会返回完整的好友所组成的列表，其中每个好友为一个字典，列表的第 1 项为本人的账号信息，如果传入 update 参数为 True，可以更新好友列表并返回。

下面是某个好友的字典信息：

```
{'OwnerUin': 0, 'AppAccountFlag': 0, 'DisplayName': '', 'KeyWord': '',
'IsOwner': 0, 'EncryChatRoomId': '', 'NickName': '富兰克林', 'UniFriend': 0,
'ContactFlag': 3, 'Province': '上海', 'RemarkPYInitial': 'FLKLFBK',
'UserName': '@bef3be95365d187525526e8f4a185cb0d06de5385d8c2b6d9a705ed39
e691c88', 'HeadImgUrl': '/cgi-bin/mmwebwx-bin/webwxgeticon?seq=636140456&
username=@bef3be95365d187525526e8f4a185cb0d06de5385d8c2b6d9a705ed39e691
c88&skey=@crypt 4a30791b 8487e5a117a9ec8b721ccfd17fe7da2f', 'Signature':
'范本恺', 'PYQuanPin': 'fulankelin', 'Sex': 1, 'SnsFlag': 49, 'AttrStatus':
33788221, 'MemberCount': 0, 'VerifyFlag': 0, 'RemarkName': '富兰克林范本恺',
'PYInitial': 'FLKL', 'RemarkPYQuanPin': 'fulankelinfanbenkai', 'City': '',
'Uin': 0, 'MemberList': <ContactList: []>, 'Alias': '', 'StarFriend': 0,
'ChatRoomId': 0, 'Statues': 0, 'HideInputBarFlag': 0}
```

从中可以得到好友的省份（'Province'）、用户 ID（'UserName'）、性别（'Sex'，值 1 表示男）等信息。

好友的搜索方法为 search_friends()，其有 4 种搜索方法：

1）仅获取自己的用户信息

```
itchat.search_friends()            #获取自己的用户信息，返回自己的属性字典
```

2）获取特定 UserName 的用户信息

```
#获取特定 UserName 的用户信息
itchat.search_friends(userName='@abcdefg1234567')
```

3）获取备注、微信号、昵称中的任何一项等于 name 键值的用户

```
#获取任何一项等于 name 键值的用户
itchat.search_friends(name='littlecodersh')
```

4）获取备注、微信号、昵称分别等于相应键值的用户

```
#获取分别对应相应键值的用户
itchat.search_friends(wechatAccount='littlecodersh')
```

第 3 种和第 4 种方法可以一起使用，下面是示例程序：

```
itchat.search_friends(name='LittleCoder机器人', wechatAccount='littlecodersh')
```

❷ 公众号

公众号的获取方法为 get_mps()，将会返回完整的公众号列表，其中每个公众号为一个字典，传入 update 参数为 True 将可以更新公众号列表并返回。

公众号的搜索方法为 search_mps()，其有两种搜索方法：

1）获取特定 UserName 的公众号

```
#获取特定 UserName 的公众号，返回值为一个字典
itchat.search_mps(userName='@abcdefg1234567')
```

2）获取名字中含有特定字符的公众号

```
#获取名字中含有特定字符的公众号，返回值为一个字典的列表
itcaht.search_mps(name='LittleCoder')
```

如果两项都做了特定，将会仅返回特定 UserName 的公众号，下面是示例程序：

```
#以下方法相当于仅特定了 UserName
itchat.search_mps(userName='@abcdefg1234567', name='LittleCoder')
```

❸ 群聊

群聊的获取方法为 get_chatrooms()，将会返回完整的群聊列表，其中每个群聊为一个字典，传入 update 参数为 True 将可以更新群聊列表并返回。

群聊的搜索方法为 search_chatrooms()，共有两种搜索方法：

1）获取特定 UserName 的群聊

```
#获取特定 UserName 的群聊，返回值为一个字典
itchat.search_chatrooms(userName='@abcdefg1234567')
```

2）获取名字中含有特定字符的群聊

```
#获取名字中含有特定字符的群聊，返回值为一个字典的列表
itcaht.search_chatrooms(name='LittleCoder')
```

如果两项都做了特定，将会仅返回特定 UserName 的群聊，下面是示例程序：

```
#以下方法相当于仅特定了 UserName
itchat.search_chatrooms(userName='@abcdefg1234567', name='LittleCoder')
```

群聊用户列表的获取方法为 update_chatroom()，群聊在首次获取中不会获取群聊的用户列表，所以需要调用该方法才能获取群聊的成员，该方法需要传入群聊的 UserName，返回特定群聊的用户列表。

```
memberList=itchat.update_chatroom('@abcdefg1234567')
```

创建群聊以及增加、删除群聊用户的方法如下，目前这 3 个方法都被严格限制了使用频率，删除群聊需要本账号为群管理员，否则会失败。

```
memberList=itchat.get_friends()[1:]
#创建群聊，topic 键值为群聊名
chatroomUserName=itchat.create chatroom(memberList, 'test chatroom')
#删除群聊内的用户
itchat.delete member from chatroom(chatroomUserName, memberList[0])
#增加用户进入群聊
itchat.add_member_into_chatroom(chatroomUserName, memberList[0])
```

7.3.6　itchat 的一些简单应用

❶ 统计微信好友的男女比例

如果想统计一下自己微信里的好友的性别比例，很简单，获取好友列表，统计列表里的性别计数。

```
import itchat
itchat.login()
#爬取自己好友的相关信息， 返回一个好友列表
friends=itchat.get friends(update=True)[0:]
#初始化计数器，有男有女，当然可能有些人没填写性别
male=female=other=0
#friends[0]是自己的信息，所以要从 friends[1]开始
for i in friends[1:]:#遍历这个列表，列表中的第 1 位是自己，所以从"自己"之后开始计算
    sex=i["Sex"]
    if sex==1:        #1 表示男性，2 表示女性
        male+=1
    elif sex==2:
        female+=1
    else:
        other +=1
#计算朋友总数
total=len(friends[1:])
#打印出自己好友的性别比例
print("男性好友：%.2f%%"%(float(male)/total*100)+"\n"+
"女性好友：%.2f%%"%(float(female)/total*100)+"\n"+
"不明性别好友：%.2f%%"%(float(other)/total* 100))
```

这好像不够直观，有兴趣的读者可以加上可视化展示，这里用基于 Python 的可视化图像库 Matplotlib，首先安装 Matplotlib 库：

```
pip install matplotlib
```

展示比例一般使用百分比圆饼表：

```
import numpy as np
#导入 Matplotlib 库
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt
labels=['man','female','unknow']
X=[ male, female, other]
fig=plt.figure()
plt.pie(X,labels=labels,autopct='%1.2f%%')
                            #画饼图（数据，数据对应的标签，百分数保留两位小数点）
plt.title("Pie chart")
plt.show()
plt.savefig("PieChart.jpg")
```

其运行效果如图 7-3 所示。

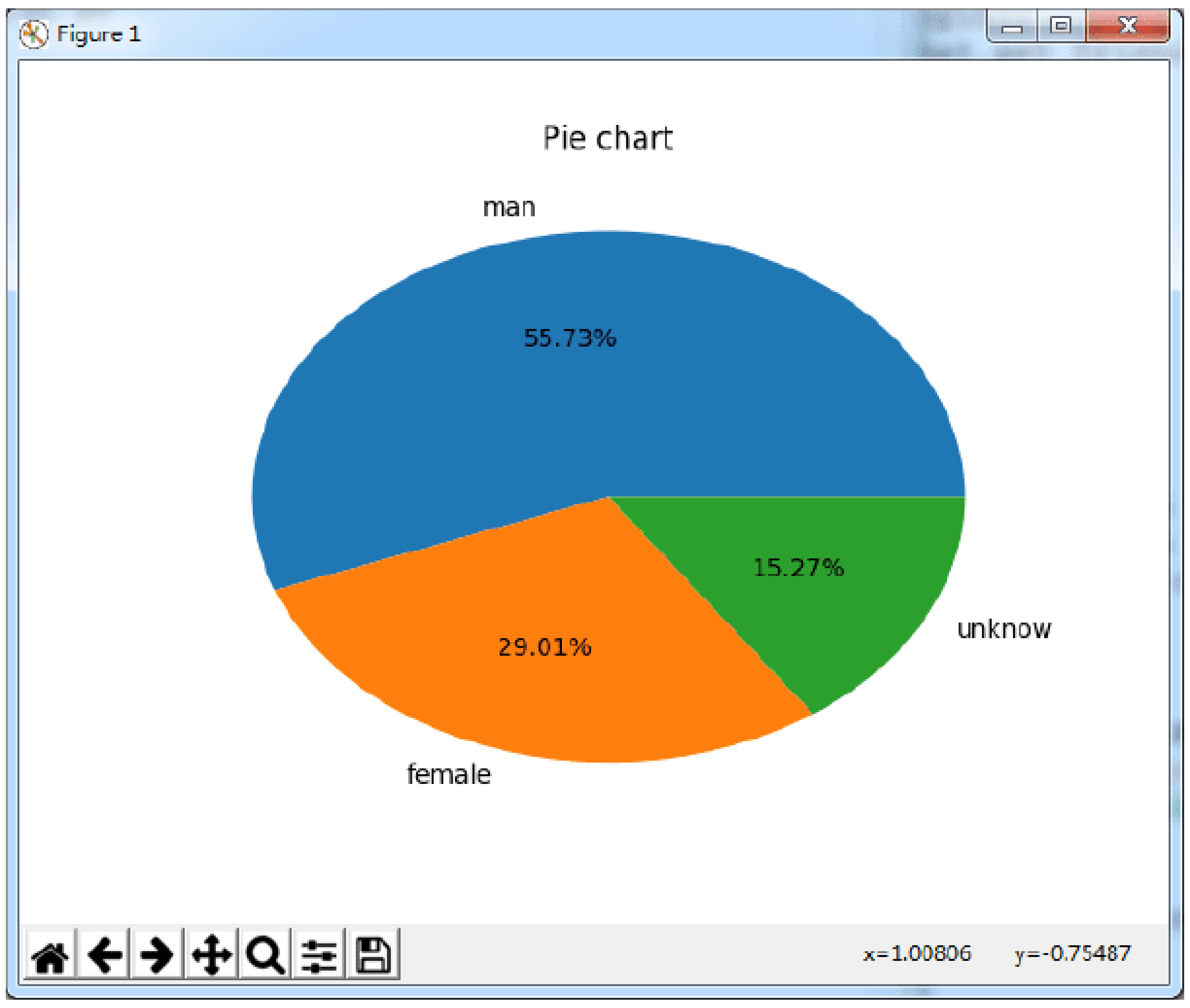


图 7-3　微信好友的男女比例饼图

❷ 统计微信好友的所在省份信息到 Excel 文件中

```
def get var(var):
    variable=[]
    for i in friends:
        value=i[var]
        variable.append(value)
    return variable
#调用函数得到各变量，并把数据存到 csv 文件中
NickName=get var("NickName")
Sex=get var('Sex')
Province=get var('Province')
City=get_var('City')
```

```
Signature=get_var('Signature')
from pandas import DataFrame
data={'NickName': NickName, 'Sex': Sex, 'Province': Province,
      'City': City, 'Signature': Signature}
frame=DataFrame(data)
frame.to_csv('data.csv', index=True)
```

❸ 微信自动回复

这里实现一个类似 QQ 上的自动回复，原理是接收到消息就发消息回去，同时发一条给文件助手，这样就可以在文件助手中统一查看消息。其代码很简单，如下：

```
#微信自动回复
import itchat
from itchat.content import *
#封装好的装饰器，当接收到的消息是 Text（即文字消息）时
@itchat.msg_register('Text')
def text_reply(msg):
    #当消息不是由自己发出的时候
    if not msg['FromUserName']==myUserName:
        #发送一条提示给文件助手
        itchat.send_msg(u"[%s]收到好友@%s 的信息：%s\n" %
          (time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(msg['CreateTime'])),
                     msg['User']['NickName'],msg['Text']), 'filehelper')
        #回复给好友
        return u'[自动回复]我现在不在,一会再和您联系。\n 已经收到您的信息:%s\n'%(msg['Text'])

if __name__=='__main__':
    itchat.auto_login()
    #获取自己的 UserName
    myUserName=itchat.get_friends(update=True)[0]["UserName"]
    itchat.run()
```

❹ 收到红包提醒

```
import itchat
from itchat.content import *
#微信红包提醒
@itchat.msg_register(NOTE,isGroupChat=True)     #监听群内红包消息 NOTE
def receive_red_packet(msg):
    if u"收到红包" in msg['Content']:
        groups=itchat.get_chatrooms(update=True)
        users=itchat.search_chatrooms(name='Happy 一家人')#把红包消息通知给这个群
        userName=users[0]['UserName']                 #获取这个群的唯一标识 ID
        for g in groups:
            if msg['FromUserName'] == g['UserName']:
                                  #根据群消息的 FromUserName 匹配是哪个群
              group_name=g['NickName']                #群的昵称
```

```
            msgbody='有人在群"%s"发了红包,请立即打电话给我,让我去抢'%group_name
            #提醒自己有人在群里发红包
            itchat.send(msgbody,toUserName=myUserName)
            #把红包消息通知给'Happy 一家人'群
            itchat.send(msgbody,toUserName=userName)
    itchat.auto_login(True)
    #获取自己的 UserName
    myUserName=itchat.get_friends(update=True)[0]["UserName"]
    itchat.run()
```

7.3.7 Python 调用图灵机器人 API 实现简单的人机交互

视频讲解

图灵机器人是一个中文语境下的对话机器人，免费的机器人允许每天有 5000 次的调用，如果放在群聊中是完全够用的（只有@的消息才使用机器人回复）。图灵机器人还有一些简单的能力，例如讲笑话、故事大全、成语接龙、新闻资讯等。如果有好友发送“讲个笑话吧”，它就会自动回复一个笑话；如果好友问“郑州天气？”，它就会自动回复天气情况，等等。

下面介绍如何简单地调用图灵机器人接口（API）。

❶ 注册获取 API KEY

这一步很简单，在“http://www.tuling123.com”对应的网页上直接注册一个账号，就可以得到自己的机器人 API KEY。这个 API KEY 在以后发送 GET 请求的时候需要用到。如果用户觉得很麻烦，也可以暂时使用 itchat 提供的几个 KEY。

```
8edce3ce905a4c1dbb965e6b35c3834d
eb720a8970964f3f855d863d24406576
1107d5601866433dba9599fac1bc0083
71f28bf79c820df10d39b4074345ef8c
```

❷ 安装 requests 实现 HTTP 请求

在安装的时候使用 pip 即可。

```
pip install requests
```

❸ 调用图灵机器人接口

其调用也比较简单，主要是模拟 POST 请求，然后解析返回的 JSON 数据。用户可以使用 requests，也可以使用 urllib 库，但 request 简化了发送 HTTP 请求的步骤。

```
import requests
import urllib
import json
KEY="e5ccc9c7c8834ec3b08940e290ff1559"    #换成自己的 API KEY
url='http://www.tuling123.com/openapi/api'
req_info='讲个笑话'.encode('utf-8')
```

```
query={'key': KEY, 'info': req info}
headers={'Content-type': 'text/html', 'charset': 'utf-8'}
```

方法一：用 requests 模块以 GET 方式获取回复内容。

```
req=requests.get(url, params=query, headers=headers)
response=req.text
data=json.loads(response)
print(data.get('text'))        #或者 data['text']
```

方法二：用 urllib 库获取回复内容。

```
data=parse.urlencode(query).encode('utf-8')#使用 urlencode()方法转换标准格式
page=request.urlopen(url,data)
html=page.read()
html=html.decode("utf-8")        #decode()将网页的信息进行解码，否则会出现乱码
data=json.loads(html)            #JSON 字典数据
print(data)                      #显示字典数据
print('机器人说: '+ data.get('text'))          #显示对话内容
```

下面是实现在 Python 的控制台下与图灵机器人聊天的例子：

```
import urllib,json
from urllib import request
from urllib import parse
def getHtml(url, data):
   page=request.urlopen(url,data)
   html=page.read()
   html=html.decode("utf-8")    #decode()将网页的信息进行解码，否则会出现乱码
   return html
if __name__=='__main__':
   key='8b005db5f57556fb96dfd98fbccfab84'
   #url='http://www.tuling123.com/openapi/api?key=' + key + '&info='+ info
   url='http://www.tuling123.com/openapi/api'
   while True:
      req_info=input('我: ')
      #发给服务器数据
      query={'key': key, 'info': req info}
      data=parse.urlencode(query).encode('utf-8')
                                          #使用 urlencode()方法转换标准格式
      response=getHtml(url, data)
      data=json.loads(response)                  #字典数据
      print(data)                                #显示字典数据
      print('机器人: '+data['text'] )            #显示对话内容
```

运行结果如下：

```
我: he
{'code': 100000, 'text': '他'}
```

```
机器人说：他
我：讲个笑话
{'code': 100000, 'text': '北京的城管也是蛮拼的。夜里十二点多，路边买个饼吃，钱都交了，卖饼的阿姨被城管吓跑了'}
机器人说：北京的城管也是蛮拼的。夜里十二点多，路边买个饼吃，钱都交了，卖饼的阿姨被城管吓跑了
```

7.4 程序设计的步骤

读者掌握了以上关键技术，就可以轻松开发微信机器人了。

```
#加载库
from itchat.content import *
import requests
import json
import itchat
itchat.auto_login()
```

调用图灵机器人的 API，采用爬虫的原理，根据聊天消息返回回复内容。

```
def tuling(info):
  appkey="e5ccc9c7c8834ec3b08940e290ff1559"
  url="http://www.tuling123.com/openapi/api?key=%s&info=%s"%(appkey,info)
  req=requests.get(url)
  content=req.text
  data=json.loads(content)
  answer=data['text']
  return answer
```

对于群聊信息，定义获取想要针对某个群进行机器人回复的群 ID 函数。

```
def group id(name):
  df=itchat.search chatrooms(name=name)
  return df[0]['UserName']
```

注册 itchat 文本消息，绑定到 text_reply()处理函数。

```
#text reply msg files 可以处理好友之间的聊天回复
@itchat.msg register([TEXT,MAP,CARD,NOTE,SHARING])
def text reply(msg):
  itchat.send('%s' % tuling(msg['Text']),msg['FromUserName'])
```

注册多媒体消息，绑定到 download_files()处理函数。

```
@itchat.msg register([PICTURE, RECORDING, ATTACHMENT, VIDEO])
def download files(msg):
  msg['Text'](msg['FileName'])
  return '@%s@%s' % ({'Picture': 'img', 'Video': 'vid'}.get(msg['Type'],
```

```
    'fil'), msg['FileName'])
```

现在微信用户通常加了很多群，但并不想对所有的群都设置微信机器人，只想对某些群设置微信机器人，可进行如下设置：

```
@itchat.msg_register(TEXT, isGroupChat=True)
def group_text_reply(msg):
  #如果只想对@自己的人回复，可以设置if msg['isAt']
  item=group_id(u'想要设置的群的名称') #根据自己的需求设置
  if msg['FromUserName']==item:
    itchat.send(u'%s' % tuling(msg['Text']), item)
itchat.run()
```

这个机器人会自动回复好友和群聊信息，并将发来的图片等多媒体信息下载下来重新发给对方。

7.5 开发消息同步机器人

有了开发微信聊天机器人的经验之后，接下来开发微信消息同步机器人，微信消息同步机器人用于完成两个群信息的同步（当任意一个群收到消息时同步到其他另一个群）。

其开发思路是设计一个字典 groups，用来存放需要同步消息的群聊的 ID，其中 key 为群聊的 ID，value 为群聊的名称。

```
groups={'群聊的 ID': 群聊的名称, '群聊的 ID ': 群聊的名称}
```

例如：

```
groups={'@f47fcf4533413b5fad998e30459a86866623c68cb6a363c7aeff208ec03fb
b8d','Happy 一家人',@f47fcf4533413b5fad998e30459a86866623c68cb6a363c7aeff
208ec03fbb8d, '神聊谷'}
```

当接收到群聊消息时，如果消息来自于需要同步消息的群聊，就根据消息类型进行处理，同时转发到其他需要同步的群聊。

首先定义一个消息响应函数，文本类消息可以用 TEXT 和 SHARING 两类，使用 isGroupChat=True 指定消息来自于群聊，这个参数默认为 False。

```
import itchat
from itchat.content import *
@itchat.msg_register([TEXT, SHARING], isGroupChat=True)
def group_reply_text(msg):
  #获取群聊的 ID，即消息来自于哪个群聊
  source=msg['FromUserName']              #群聊的 ID
  #这里可以把 source 打印出来，在确定是哪个群聊后把群聊的 ID 和名称加入 groups
  groups={'@f47fcf4533413b5fad998e30459a86866623c68cb6a363c7aeff208ec03
  fbb8d',' Happy 一家人',@f47fcf4533413b5fad998e30459a86866623c68cb6a363c7
  aeff208ec03fbb8d, '神聊谷'}
```

```
    #处理文本消息
    if msg['Type']==TEXT:
      #消息来自于需要同步消息的群聊
      if source in groups:#判断是否在字典里，等价于2.7版本中的groups.has_key(source)
        for item in groups.keys():      #转发到其他需要同步消息的群聊
          if not item==source:
            #groups[source]: 消息来自于哪个群聊
            #msg['ActualNickName']: 发送者的名称
            #msg['Content']: 文本消息内容
            #item: 需要被转发的群聊 ID
            itchat.send('%s: %s\n%s' % (groups[source], msg['ActualNickName'],
            msg['Content']), item)
    #处理分享消息
    elif msg['Type']==SHARING:
      if groups.has_key(source):
        for item in groups.keys():
          if not item==source:
            #msg['Text']: 分享的标题，msg['Url']: 分享的链接
            itchat.send('%s: %s\n%s\n%s' % (groups[source], msg['ActualNickName'],
            msg['Text'], msg['Url']), item)
```

再来处理一下图片等多媒体类消息。

```
#处理图片和视频类消息
@itchat.msg_register([PICTURE, VIDEO], isGroupChat=True)
def group_reply_media(msg):
  source=msg['FromUserName']
  #下载图片或视频
  msg['Text'](msg['FileName'])
  if groups.has_key(source):
    for item in groups.keys():
      if not item==source:
        #将图片或视频发送到其他需要同步消息的群聊
        itchat.send('@%s@%s' % ({'Picture': 'img', 'Video': 'vid'}.get(msg['Type'],
        'fil'), msg['FileName']), item)
itchat.auto_login(True)
itchat.run()
```

以上代码实现了对文本、分享、图片、视频 4 类消息的处理，如果用户对其他类型的消息也感兴趣，进行相应的处理即可。目前两个群之间可以进行消息的同步了，一群和二群的用户之间终于可以畅快地聊起来。

第 8 章

微信网页版协议应用——微信机器人

视频讲解

8.1 微信网页版机器人功能介绍

微信网页版是腾讯公司开发的微信官方工具，能够在计算机网页上使用微信，微信网页版基于 Https 请求，通过 API 的形式与微信服务器进行数据交互，所有的协议均暴露，可通过浏览器抓包工具进行获取和分析。

目前国内流行的浏览器（例如 Chrome 浏览器、FireFox 浏览器）均带有开发者工具，用户通过网络请求面板可以详细地看到整个微信网页版在运行过程中涉及的所有 API 请求。本章根据上述获取到的请求模拟微信网页版的流程。

首先对微信网页版的整体运行流程进行分析，使用 Python 语言模拟微信网页版的运行流程，实现登录、获取好友信息、发送消息等基础操作，并在上述基础之上实现一些扩展操作，例如自动确认好友请求、定时发送消息、检测好友状态、自动邀请好友加入群聊等。

通过对微信网页版协议进行分析，读者对 API 中的请求参数、请求方式、返回值等加深了理解，这对后续的实际开发有很多值得借鉴的地方；能够清晰地了解微信网页版的整体运行流程，扩展自己的思维。另外，在扩展个人微信号的同时也能很好地练习 Python 基础，这对学习网络爬虫等有很大的帮助意义。

8.2 微信网页版机器人设计思路

8.2.1 分析微信网页版 API

本章以 Chrome 浏览器为例讲解如何抓包分析微信网页版 API，步骤如下：

❶ 打开开发者工具

首先打开 Chrome 浏览器，然后打开开发者工具，如图 8-1 所示。

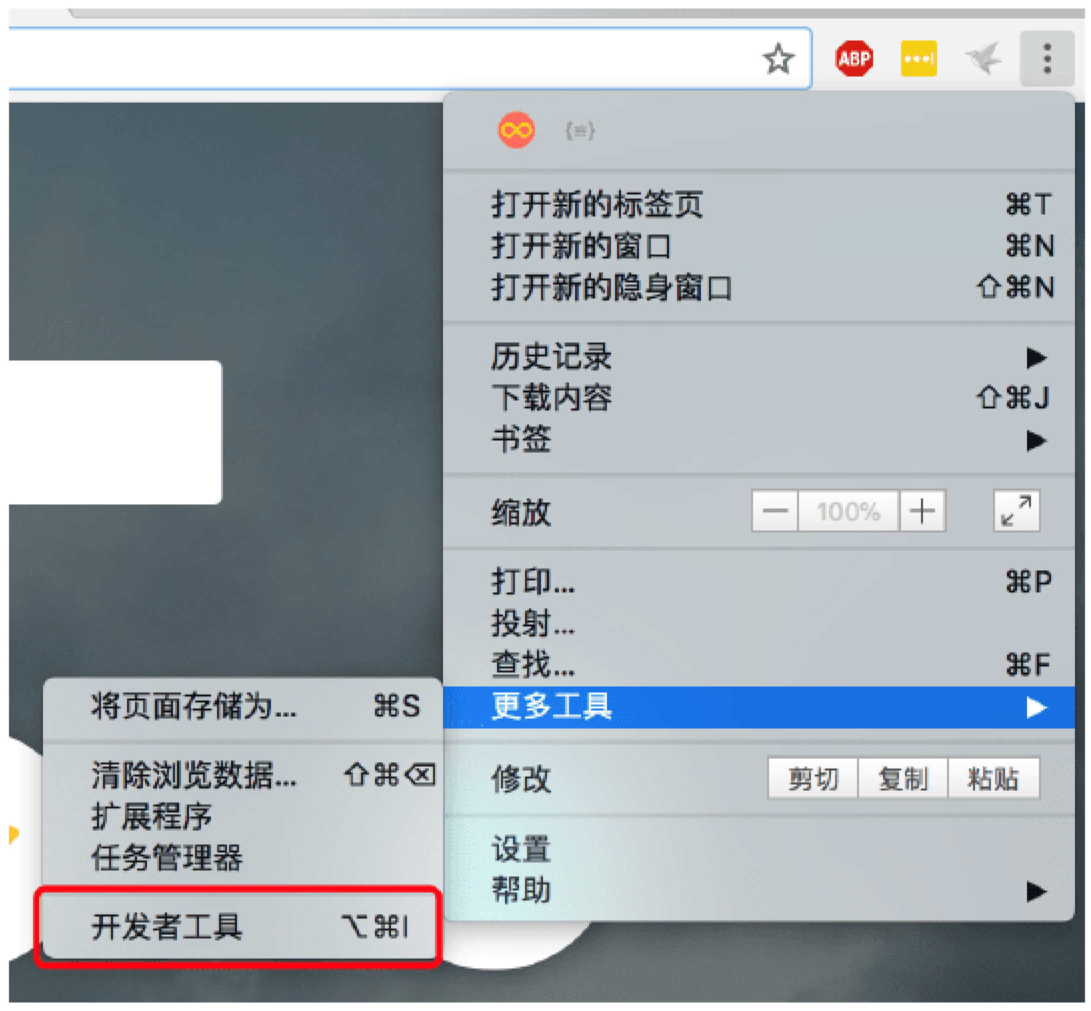


图 8-1　打开开发者工具

切换到 Network 界面，如图 8-2 所示。

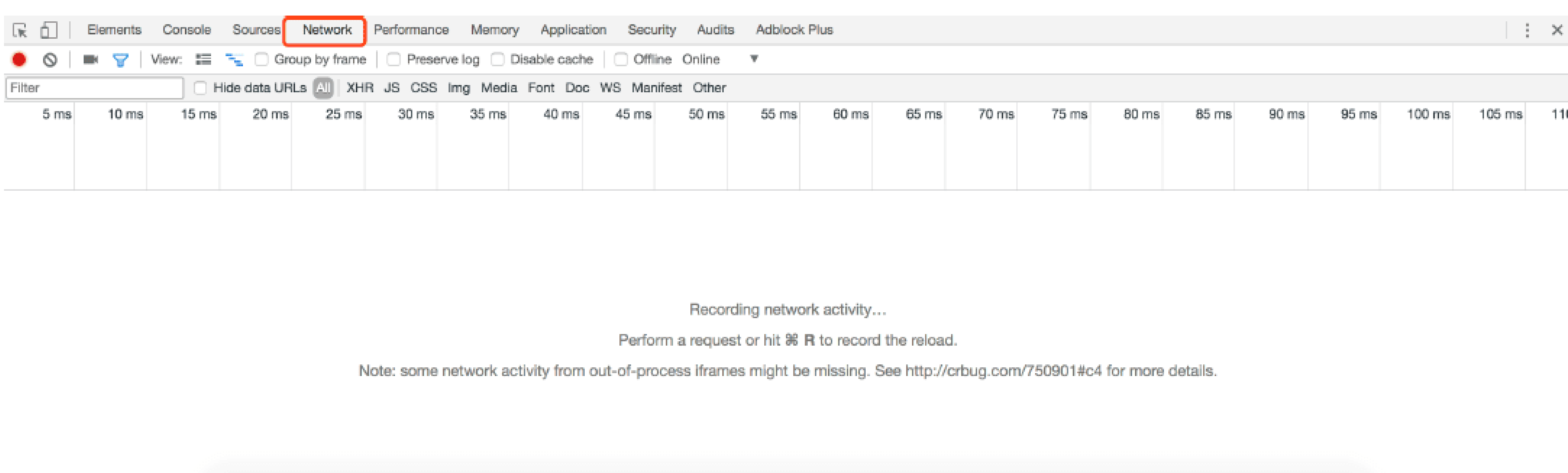


图 8-2　切换到 Network 界面

勾选 Preserve log 复选框，否则在跳转之后无法查看之前的请求信息。

提示：对于 Chrome 开发者工具的使用小技巧，读者可以在网址“http://blog.csdn.net/Letasian/article/details/78461438”对应的页面中查看。

❷ 打开微信网页版

在浏览器中输入网址（https://wx.qq.com），打开微信网页版。在左侧 URL 列表中可以看到以 jslogin 开始的网址，该 URL 即为获取登录所需二维码的 API 请求。

在图 8-3 中，左侧为所有的请求，单击某一个请求后右侧会出现该请求的详细信息。

General 中为请求的 URL（Request URL）和请求方式（Request Method），常见的请求方式有 GET 和 POST。Response Headers 为响应头信息，Request Headers 为请求头信息，包括常用的 User-Agent、Host、Referer、Cookie 等，Query String Parameters 为请求 URL 中的参数信息。

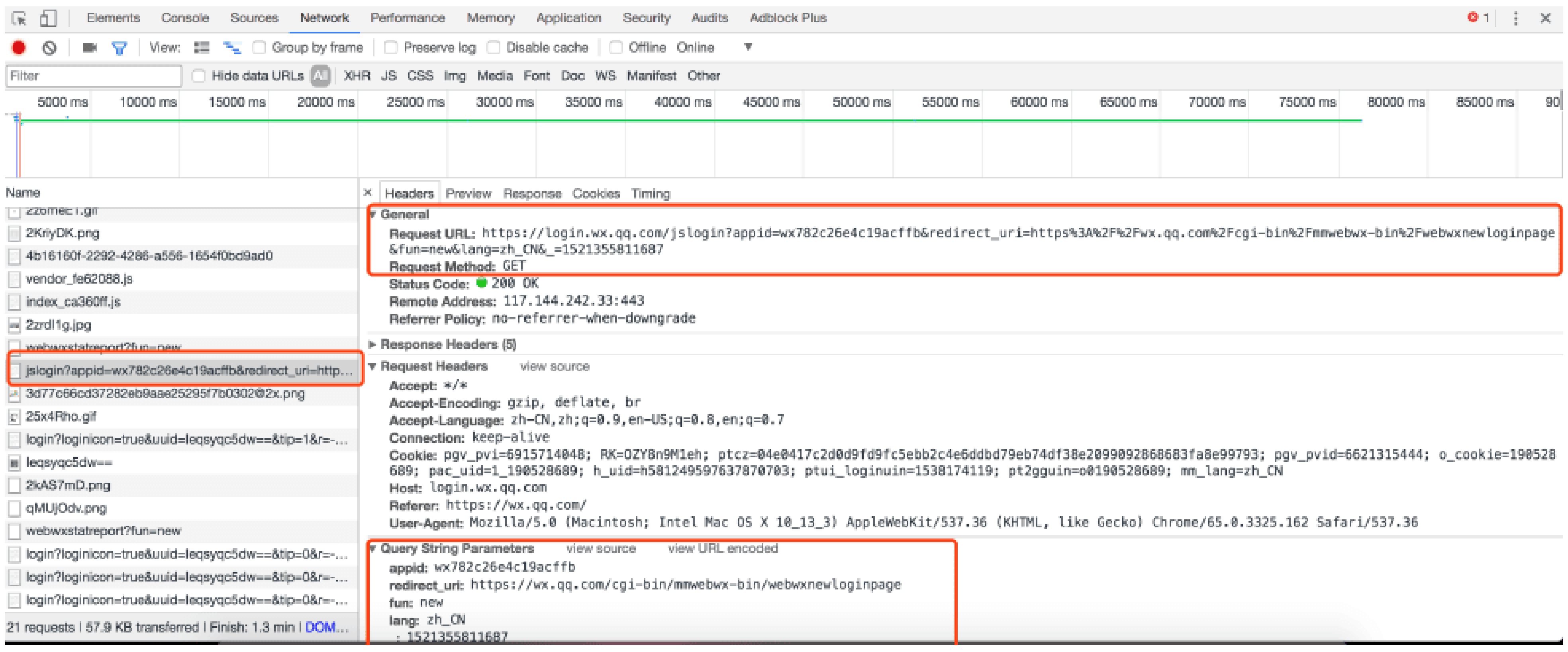


图 8-3 分析 URL

单击右侧的 Response 标签，可以看到格式如“window.QRLogin.code=200; window.QRLogin.uuid=" QZNW7IsPLg== ";”的内容，其中 window.QRLogin.code 代表该请求发送成功，window.QRLogin.uuid 中的内容为当前登录所需要的二维码信息，将该 uuid 传入获取二维码的 URL 中即可获取到二维码的图片。

❸ API 总结

根据上述过程可以得知该 URL（https://login.wx.qq.com/jslogin?appid=wx782c26e4c19acffb&redirect_uri=https%3A%2F%2Fwx.qq.com%2Fcgi-bin%2Fmmwebwx-bin%2Fwebwxnewloginpage&fun=new&lang=zh_CN&_=1521355811687）即为获取登录所需二维码的 API，请求方式为 GET，其中参数列表中的 appid 为微信开放平台注册的应用的 AppID，“_”为当前时间的 13 位毫秒值。

其余参数均固定，如下。

- redirect_uri：https://wx.qq.com/cgi-bin/mmwebwx-bin/webwxnewloginpage。
- fun：new。
- lang：zh_CN。

注：appid 就是在微信开放平台注册的应用的 AppID。网页版微信有两个 AppID，早期的是 wx782c26e4c19acffb，在微信客户端上显示的应用名称为 Web 微信；现在用的是 wxeb7ec651dd0aefa9，显示的名称为微信网页版。

根据上述分析，可以将此 API 归纳为表 8-1。

表 8-1 获取登录所需的二维码

描　述	获取登录所需的二维码
地址	https://login.wx.qq.com/jslogin
请求类型	GET
请求参数	appid：wx782c26e4c19acffb redirect_uri：https://wx.qq.com/cgi-bin/mmwebwx-bin/webwxnewloginpage fun：new lang：zh_CN _：时间戳
返回值	window.QRLogin.code=200; window.QRLogin.uuid="oZOD_53KKw==";

其余 API 的分析方法与之类似，此处不再一一介绍。

8.2.2 API 汇总

下面 API 列表中的参数均为举例，在实际运行中很多都是动态的，需要用户自定义传入，返回值也为样例，对于部分过长的有删减，不再一一指出。

❶ 显示二维码

显示二维码如表 8-2 所示。

表 8-2 显示二维码

描　　述	显示二维码
地址	https://login.weixin.qq.com/qrcode/{uuid}
请求类型	GET
请求参数	无
返回值	图片二进制流

❷ 等待扫描二维码

等待扫描二维码如表 8-3 所示。

表 8-3 等待扫描二维码

描　　述	等待微信手机客户端扫描二维码
地址	https://login.wx.qq.com/cgi-bin/mmwebwx-bin/login
请求类型	GET
请求参数	loginicon：true uuid：xxxx tip：0 为未扫描，1 为已扫描 r：(−940126109)，毫秒值取反 _：1494054830403，时间戳
返回值	window.code=xxx; xxx：408 表示登录超时；201 表示扫描成功；200 表示确认登录 当 code 为 200 时数据包括跳转地址，例如“window.redirect_uri="https://wx2.qq.com/cgi-bin/mmwebwx-bin/webwxnewloginpage?ticket=AwglwtXTXfw9oQgP9js1V1ig@qrticket_0&uuid=4fVCtDVBOg==&lang=zh_CN&scan=1494055480";”，需要将 ticket 参数保存

❸ 登录获取 Cookie

登录获取 Cookie 如表 8-4 所示。

表 8-4 登录获取 Cookie

描　　述	登录成功后获取 Cookie 信息
地址	https://wx2.qq.com/cgi-bin/mmwebwx-bin/webwxnewloginpage
请求类型	GET
请求参数	ticket：AwglwtXTXfw9oQgP9js1V1ig@qrticket_0 uuid：4fVCtDVBOg== lang：zh_CN scan：1494055480 fun：new version：v2

续表

描　　述	登录成功后获取 Cookie 信息
返回值	<error> <ret>0</ret> <message></message> <skey>@crypt_522d8a57_60a7646e99d8f25b230</skey> <wxsid>VOXSzwU5lNcrbor8</wxsid> <wxuin>2929094227</wxuin> <pass_ticket>aGK4HWSoVzFoQscFt2hO%2Fy</pass_ticket> <isgrayscale>1</isgrayscale> </error>

❹ 微信初始化

微信初始化如表 8-5 所示。

表 8-5　微信初始化

描　　述	微信初始化
地址	https://wx2.qq.com/cgi-bin/mmwebwx-bin/webwxinit
请求类型	POST
数据类型	JSON
请求头	Content-Type: application/json; charset=UTF-8

请求参数	{
	BaseRequest: { Uin: xxx, Sid: xxx, Skey: xxx, DeviceID: xxx }
	}

返回值	{
	"BaseResponse": { "Ret": 0, "ErrMsg": "" },
	"Count": 11,
	"ContactList": [...],
	"SyncKey": {
	"Count": 4,
	"List": [
	{
	"Key": 1,
	"Val": 635705559
	},
	...
	]
	},
	"User": {
	"Uin": xxx,
	"UserName": xxx,
	"NickName": xxx,
	…
	},
	"ChatSet": xxx,
	"SKey": xxx,

续表

描　述	微信初始化
返回值	"ClientVersion": 369297683, "SystemTime": 1453124908, "InviteStartCount": 40, "MPSubscribeMsgCount": 2, "MPSubscribeMsgList": [...], "ClickReportInterval": 600000 }

❺ 开启手机状态通知

开启手机状态通知如表 8-6 所示。

表 8-6　开启手机状态通知

描　述	开启手机微信客户端状态通知
地址	https://wx2.qq.com/cgi-bin/mmwebwx-bin/webwxstatusnotify
请求类型	POST
参数类型	JSON
请求头	Content-Type: application/json; charset=UTF-8

请求参数	{
	BaseRequest：{ Uin: xxx, Sid: xxx, Skey: xxx, DeviceID: xxx },
	Code: 3,
	FromUserName: 自己的 ID,
	ToUserName: 自己的 ID,
	ClientMsgId: 时间戳
	}

返回数据	{
	"BaseResponse": {"Ret": 0,"ErrMsg": ""},
	"MsgID": "1848761205298770623"
	}

❻ 获取联系人列表

获取联系人列表如表 8-7 所示。

表 8-7　获取联系人列表

描　述	获取联系人列表
地址	https://wx2.qq.com/cgi-bin/mmwebwx-bin/webwxgetcontact
请求类型	POST
参数类型	JSON
请求头	Content-Type: application/json; charset=UTF-8

请求参数	{
	BaseRequest: { Uin: xxx, Sid: xxx, Skey: xxx, DeviceID: xxx}
	}

返回数据	{
	"BaseResponse": {
	"Ret": 0,
	"ErrMsg": ""
	},
	"MemberCount": 334,
	"MemberList": [
	{
	"Uin": 0,

续表

描　　述	获取联系人列表
返回数据	"UserName": xxx, "NickName": "Urinx", … }, …], "Seq": 0 }

❼ 获取指定联系人详细信息

获取指定联系人详细信息如表 8-8 所示。

表 8-8　获取指定联系人详细信息

描　　述	获取指定联系人详细信息
地址	https://wx2.qq.com/cgi-bin/mmwebwx-bin/webwxbatchgetcontact
请求类型	POST
参数类型	JSON
请求头	Content-Type: application/json; charset=UTF-8

请求参数	{"BaseRequest": {
	"Uin": 2929094227,
	"Sid": "VOXSzwU5lNcrbor8",
	"Skey": "@crypt_522d8a57_68f25b230",
	"DeviceID": "e586776133027541"
	},
	"Count": 1,
	"List": [
	{
	"UserName": "@591768165ac0c9b8326bbe1355",
	"EncryChatRoomId": "@@62ea4eba9ecd4a8111327b7cb"
	}
	]
	}

返回数据	{
	"BaseResponse": {
	"Ret": 0,
	"ErrMsg": ""
	},
	"Count": 1,
	"ContactList": [
	{
	"Uin": 0,
	"UserName": "@59176816542bb0c9b8326bbe1355",
	"NickName": "路人甲",
	"HeadImgUrl": "/cgi-bin/mmwebwx-bin/webwxgeticon?seq=0&username=@591755&chatroomid=@6a8f2&skey=",
	"ContactFlag": 0,
	…
	}
	]
	}

❽ 检查新消息

检查新消息如表 8-9 所示。

表 8-9　检查新消息

描　　述	检查新消息
地址	https://webpush2.weixin.qq.com/cgi-bin/mmwebwx-bin/synccheck
请求类型	GET
参数类型	JSON
请求头	Content-Type: application/json; charset=UTF-8

请求参数	{
	　　BaseRequest: { Uin: xxx, Sid: xxx, Skey: xxx, DeviceID: xxx }
	}

返回值	window.synccheck={retcode:"xxx",selector:"xxx"} retcode：0 为正常；1100 表示失败/退出微信；1101 表示在其他地方登录 Web 版微信；1102 表示在手机上主动退出 selector：0 为正常；2 表示新的消息；6 表示联系人信息变更；7 表示进入/离开聊天界面

❾ 获取新消息

获取新消息如表 8-10 所示。

表 8-10　获取新消息

描　　述	获取新消息
地址	https://wx2.qq.com/cgi-bin/mmwebwx-bin/webwxsync
请求类型	POST
参数类型	JSON
请求头	Content-Type: application/json; charset=UTF-8

请求参数	{
	　　BaseRequest: { Uin: xxx, Sid: xxx, Skey: xxx, DeviceID: xxx },
	　　SyncKey: xxx,
	　　rr: 时间戳取反
	}

返回值	{ 'BaseResponse': {'ErrMsg': '', 'Ret': 0},
	'SyncKey': { 'Count': 7,
	　　　　'List': [{'Val': 636214192, 'Key': 1},]
	　　}, 'ContinueFlag': 0,
	　　'AddMsgCount': 1,
	　　'AddMsgList': [{
	　　　　　　'FromUserName': '',
	　　　　　　'RecommendInfo': {...},
	　　　　　　'Content': "",
	　　　　　　...
	　　　　},
	　　　　...
	　　],
	}

❿ 发送文本消息

发送文本消息如表 8-11 所示。

表 8-11 发送文本消息

描 述	发送文本消息
地址	https://wx2.qq.com/cgi-bin/mmwebwx-bin/webwxsendmsg
请求类型	POST
参数类型	JSON
请求头	Content-Type: application/json; charset=UTF-8

请求参数	{
	BaseRequest: { Uin: xxx, Sid: xxx, Skey: xxx, DeviceID: xxx },
	Msg: {
	Type: 1 文字消息,
	Content: 要发送的消息,
	FromUserName: 自己的 ID,
	ToUserName: 好友的 ID,
	LocalID: 与 clientMsgId 相同,
	ClientMsgId: 时间戳左移 4 位随后补上 4 位随机数
	}
	}

返回值	{"BaseResponse": { "Ret": 0, "ErrMsg": ""}}

⓫ 发送图片消息

发送图片消息如表 8-12 所示。

表 8-12 发送图片消息

描 述	发送图片消息
地址	https://wx2.qq.com/cgi-bin/mmwebwx-bin/webwxsendmsgimg
请求类型	POST
参数类型	JSON
请求头	Content-Type: application/json; charset=UTF-8

请求参数	{
	BaseRequest: { Uin: xxx, Sid: xxx, Skey: xxx, DeviceID: xxx },
	Msg: {
	Type: 3 图片消息,
	MediaId: 图片上传后的媒体 ID,
	FromUserName: 自己的 ID,
	ToUserName: 好友的 ID,
	LocalID: 与 clientMsgId 相同,
	ClientMsgId: 时间戳左移 4 位随后补上 4 位随机数
	}
	}

返回值	{"BaseResponse": { "Ret": 0, "ErrMsg": ""}}

注：非文本消息，例如图片、语音、视频、表情等，API 类似，均是先通过上传文件获取媒体 ID，然后发送此媒体 ID。

⓬ 获取头像

获取头像如表 8-13 所示。

表 8-13　获取头像

描　　述	获 取 头 像
地址	https://wx.qq.com/cgi-bin/mmwebwx-bin/webwxgeticon
请求类型	GET
请求参数	seq：数字 username：用户 ID skey：xxx
返回值	二进制

注：获取好友、群头像等 API 类似。

⓭ 获取图片消息

获取图片消息如表 8-14 所示。

表 8-14　获取图片消息

描　　述	获取图片消息
地址	https://wx.qq.com/cgi-bin/mmwebwx-bin/webwxgetmsgimg
请求类型	GET
请求参数	msgid：消息 ID username：用户 ID type：slave 缩略图，为空时加载原图 skey：xxx
返回值	二进制

注：获取语音、视频、表情等消息类似，无 type 参数。

8.2.3　其他说明

❶ 账号类型介绍

账号类型如表 8-15 所示。

表 8-15　账号类型

类　　型	说　　明
个人账号	以“@”开头，后面由 32 位数字和字母组成
群聊	以“@@”开头，后面由 32 位数字和字母组成
公众号/服务号	以“@”开头，但其 VerifyFlag & 8 != 0 VerifyFlag: 　　一般个人公众号/服务号：8 　　一般企业的服务号：24 　　微信官方账号微信团队：56
特殊账号	像文件传输助手之类的账号，它们有特殊的 ID，目前已知的有 filehelper、newsapp、fmessage、weibo、qqmail、tmessage、qmessage、qqsync、floatbottle、lbsapp、shakeapp、medianote、qqfriend、readerapp、blogapp、facebookapp、masssendapp、meishiapp、feedsapp、voip、blogappweixin、weixin、brandsessionholder、weixinreminder、officialaccounts、notification_messages、wxitil、userexperience_alarm、notification_messages

例如测试发送消息，可将接收人（ToUserName）指定为 filehelper（文件传输助手）进行测试。

❷ 消息类型介绍

微信中消息的格式一般如下：

```
{
    "FromUserName": "",
    "ToUserName": "",
    "Content": "",      #内容
    "StatusNotifyUserName": "",
    "ImgWidth": 0,
    "PlayLength": 0,
    "RecommendInfo": {...},
    "StatusNotifyCode": 4,
    "NewMsgId": "",
    "Status": 3,
    "VoiceLength": 0,
    "ForwardFlag": 0,
    "AppMsgType": 0,
    "Ticket": "",
    "AppInfo": {...},
    "Url": "",
    "ImgStatus": 1,
    "MsgType": 1,
    "ImgHeight": 0,
    "MediaId": "",
    "MsgId": "",
    "FileName": "",
    "HasProductId": 0,
    "FileSize": "",
    "CreateTime": 1454602196,
    "SubMsgType": 0
}
```

其中经常用到的字段有 FromUserName（发送人 ID）、ToUserName（接收人 ID）、Content（内容）、MsgId（消息 ID）、MsgType（消息类型）等。微信中的部分消息类型如表 8-16 所示。

表 8-16　微信中的消息类型

类　型	说　明
1	文本消息
3	图片消息
34	语音消息
37	好友确认消息
43	视频消息
48	位置消息
49	分享链接
10000	系统消息
10002	撤回消息

当 MsgType=1 时，用户只需要获取 Content 字段即可获取消息内容，当 MsgType=3 时，获取 MediaId 字段，将 MediaId 传入获取文件接口，即可下载此图片消息。语音、视频等消息的获取与此类似。

8.3 程序设计的步骤

8.3.1 微信网页版的运行流程

从上述微信网页版 API 的分析过程可以得出微信网页版的运行流程，如图 8-4 所示。

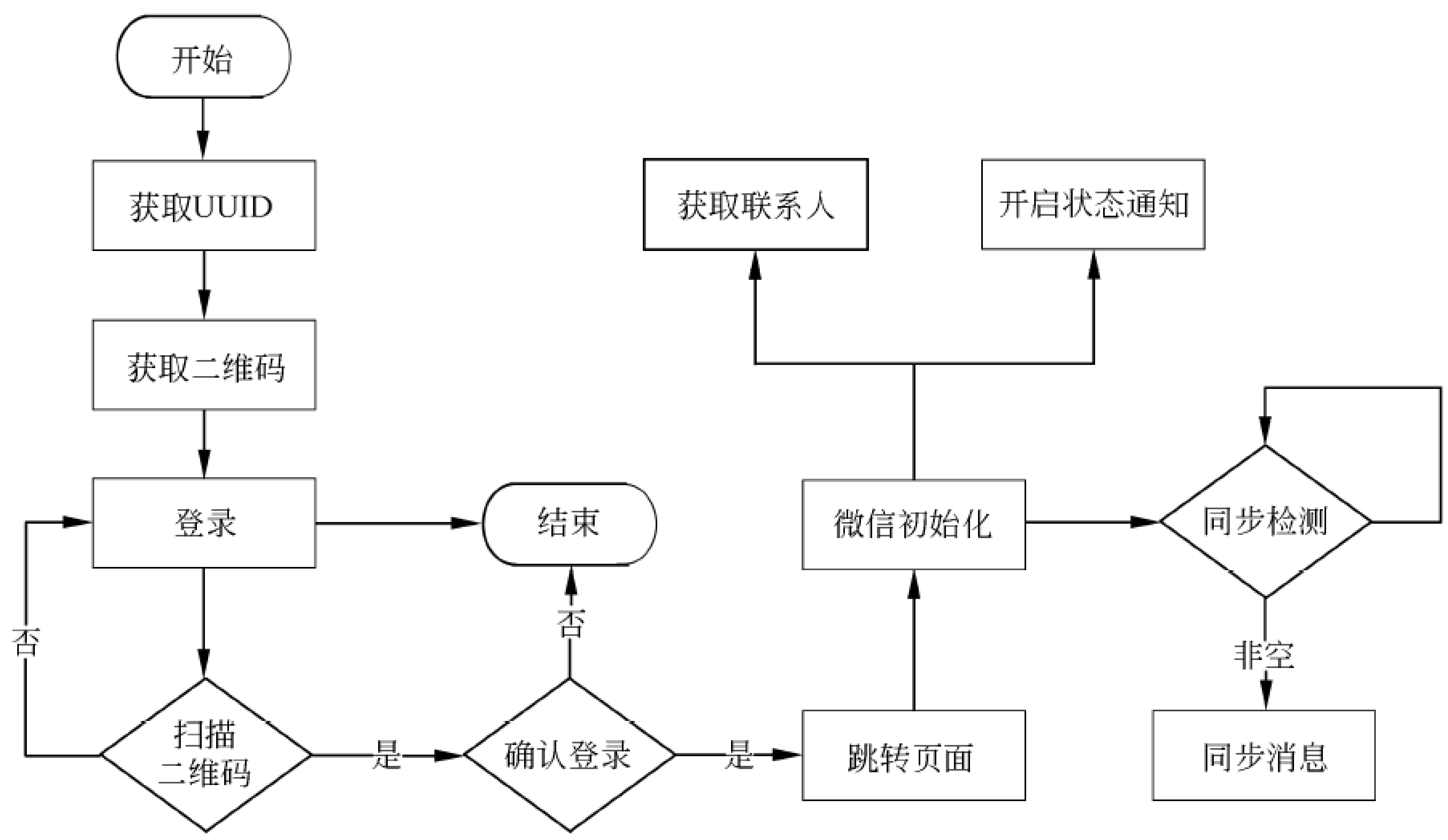


图 8-4　微信网页版的运行流程

（1）打开微信网页版（https://wx.qq.com），获取随机的 UUID。

（2）根据第 1 步获得的 UUID 获取登录所需扫描的二维码图片。

（3）使用微信移动客户端扫描该二维码，检测用户是否已经扫码。

（4）扫描二维码成功后，检测用户是否单击确认登录。

（5）确认登录后，跳转新页面，调用初始化 API。

（6）调用开启手机状态通知 API，获取联系人列表。

（7）循环执行同步检测，待收到响应后继续发起下一个请求。此时根据返回数据的内容判断是否需要拉取消息、自动退出等操作。

注：目前微信网页版更新出账号记录功能，用户在登录时可根据之前的账号信息实现免扫码，直接在微信移动客户端单击确认登录。本程序未实现此功能，感兴趣的读者可以自行实现。

8.3.2　程序目录

本系统的程序文件目录如下：

```
├── wechat_bot          项目目录
    ├── wechat_bot.py      程序入口
    ├── thread_pool  线程相关包
        ├── send_msg_thread.py   发送消息线程
    ├── wechat          微信相关包
        ├── wechat_msg_processor.py        消息处理
        ├── wechat.py          模拟微信运行类（继承 wechat_api）
        ├── wechat_apis.py     基础协议（包括所有 API 抽象的函数）
    ├── base    基础包
        ├── config_manager.py   读取配置文件
        ├── log.py        日志
        ├── utils.py       常用工具
        ├── constant.py    相关常量
        ├── wechat.conf.bak  基础配置文件
    ├── data   运行数据
        ├── 1001      机器人编号
            ├── Logs  日志
            ├── Data  文件
                ├── msgs   消息
                ├── users   好友头像
                ├── rooms  群聊头像
                ├── upload  上传文件
```

注：本程序实现了一台机器同时运行多个微信机器人，启动时使用如下命令：

```
python wechat_bot.py 1001
```

其中，1001 为机器人编号（数字类型），不同机器人的数据位于不同的目录下。

此时会根据 wechat.conf.bak 配置文件的内容复制生成 wechat_1001.conf 文件，如果遇到部分机器人需要修改配置文件的内容，只需修改对应编号的配置文件即可，不会影响其他机器人。

8.3.3　微信网页版运行代码的实现

❶ 获取随机 UUID 的代码

获取随机 UUID 的函数包括构造请求参数，使用 Request 发送 POST 请求，在返回值中使用正则表达式截取 code 和 uuid，并将 uuid 设置到变量中。

```
def getuuid(self):
  """
  获取登录需要的 UUID
  :return: True or False
  """
  try:
     url=self.wx_conf['API_jsLogin']
     params={                                      #构造请求参数
         'appid': self.appid,
         'fun': 'new',
         'lang': self.wx_conf['LANG'],
         '_': int(time.time()),
         'redirect_uri': self.wx_conf['API_jslogin_redirect_url'],
     }
     data=post(url, params, False)                 #发送 POST 请求
     regx=r'window.QRLogin.code=(\d+); window.QRLogin.uuid="(\S+?)"'
     pm=re.search(regx, data)                      #正则截取
     if pm:
         code=pm.group(1)
         self.uuid=pm.group(2)
         return code=='200'
     return False
  except:
     self.Log.error(traceback.format_exc())
```

❷ 结合 qrcode 生成二维码并实现控制台输出

使用 qrcode 包实现将二维码转化为黑白格输出在控制台，如果 IDE 为白色背景，需要将 BLACK 和 WHITE 的内容互换。

```
def genqrcode(self):
    """
    在控制台输出二维码
    :return:
    """
    str2qr_terminal(self.wx_conf['API_qrcode'] + self.uuid)
def str2qr_terminal(text):
    qr=qrcode.QRCode()
    qr.border=1
    qr.add_data(text)
    mat=qr.get_matrix()
    print_qr(mat)
def print_qr(mat):
    for i in mat:
        BLACK='\033[47m  \033[0m'
        WHITE='\033[40m  \033[0m'
        print(''.join([BLACK if j else WHITE for j in i]))
```

注：因微信网页版的运行过程涉及的流程过多，代码过长，此处不再一一列出，读者可根据下面调度函数的执行直接查看对应代码。

❸ 运行流程调度函数

```
def start(self):
    self.Log.info(Constant.LOG_MSG_START)  #记录机器人运行
    d1=datetime.now()                     #启动时间
    flag=True
    while True:
        d2=datetime.now()                 #当前时间
        d=d2 - d1
        if d.seconds > 240:               #超过 240 秒未扫码自动结束进程
            os.system(Constant.LOG_MSG_KILL_PROCESS % os.getpid())
        if flag:
            run(Constant.LOG_MSG_GET_UUID, self.getuuid)#获取二维码
            self.Log.info(Constant.LOG_MSG_GET_QRCODE)  #记录获取二维码
            self.genqrcode()              #打印二维码
            self.Log.info(Constant.LOG_MSG_SCAN_QRCODE)  #记录需要扫描二维码
            flag=False
            d1=datetime.now()
        tm=int(time.time())
        if not self.waitforlogin(tm=tm):#等待确认
            self.Log.info(time.strftime("%Y-%m-%d %H:%M:%S"))
            continue
        self.Log.info(Constant.LOG_MSG_CONFIRM_LOGIN)
        break
    run(Constant.LOG_MSG_LOGIN, self.login)           #登录
    run(Constant.LOG_MSG_INIT, self.webwxinit)        #初始化
    run(Constant.LOG_MSG_STATUS_NOTIFY, self.webwxstatusnotify)#开启状态通知
    run(Constant.LOG_MSG_GET_CONTACT, self.get_contact)    #获取联系人
    self.Log.info(Constant.LOG_MSG_CONTACT_COUNT % (
        self.MemberCount, len(self.MemberList)
    ))                                    #记录好友信息
    self.Log.info(Constant.LOG_MSG_OTHER_CONTACT_COUNT % (
        len(self.GroupList), len(self.ContactList),
        len(self.SpecialUsersList), len(self.PublicUsersList)
    ))                                    #记录群聊、特殊账号信息
    run(Constant.LOG_MSG_GET_GROUP_MEMBER,self.fetch_group_contacts)#获取群成员
    ret=self.send_text('filehelper', "我上线了...") #发送上线消息
    self.Log.info(ret)
    gc.collect()                          #回收垃圾
    self.Log.info("groups number: %d" % len(self.GroupList))
    self.init_thread()                    #初始化线程
```

```
        while True:
            [retcode, selector]=self.synccheck() #检测新消息
            self.Log.debug('tag:%s, retcode: %s, selector: %s' % (self.tag,
            retcode, selector))
            self.exit code=int(retcode)
            if retcode=='1100':
                self.Log.info(Constant.LOG MSG LOGOUT)
                break
            if retcode=='1101':
                self.Log.info(Constant.LOG MSG LOGIN OTHERWHERE)
                break
            if retcode=='1102':
                self.Log.info(Constant.LOG MSG QUIT ON PHONE)
                break
            elif retcode=='0':
                if selector=='0':
                    time.sleep(self.time out)
                else:
                    r=self.webwxsync()          #获取消息
                    if r is not None:
                        try:
                            self.handle_mod(r)    #处理联系人变更
                            self.handle_msg(r)    #处理消息
                        except Exception as ex:
                            self.Log.info(r)
                            self.Log.error(traceback.format exc())
            else:                                   #未知状态
                r=self.webwxsync()
                self.Log.debug('sync check error webwxsync: %s\n' % json.dumps(r))
```

8.4 扩展功能

8.4.1 自动回复

自动回复分为关键词回复和普通话语回复，实现逻辑是在程序启动时初始化关键词字典，其中 key 为关键词内容进行 md5 加密，value 为要回复的内容。当收到好友消息后，在关键词字典中检测是否有对应的键，若存在，则自动回复该内容；对于不在关键词中的话语，则将好友发送的话语作为内容，调用第三方机器人获取返回值进行回复。

本程序使用图灵机器人的 API 实现对话，对于图灵机器人的相关内容，请读者查看其官方文档，网址为“https://www.kancloud.cn/turing/web_api/522989”。

❶ 初始化关键词字典

```
self.keyword_reply={        #初始化关键词
     get_md5_value("你好"): "我很好，你呢？",
     get_md5_value("你是谁"): "我是可爱的 GeekBot!",
}
```

注：本程序直接初始化了字典，在实际开发中为了方便管理可使用数据库管理。

❷ 封装调用图灵机器人 API 函数

```
def call_tuling(self, text, user_id=None):
    """
    调用图灵机器人
    :param text: 消息内容
    :param user_id: 用户唯一标识，用于上下文
    :return: 回复的内容
    """
    msg={  #API 所需参数
        'key': Constant.TULING_API_KEY,  #key
        'info': text,        #内容
        'userid': user_id    #用户唯一标识
    }
    res=post(Constant.TULING_API_URL, msg)
    if res:
        if int(res['code'])==100000:
            return res['text']
        elif int(res['code'])==200000:
            return res['text'] + '\r' + res['url']
        else:
            return res['text']
    else:                        #异常
        return Constant.TULING_NOT_RES  #'我不知道你在说些什么，换个话题吧...'
```

❸ 自动回复实现

```
if get_md5_value(content) in self.wechat.keyword_reply:#判断是否存在该关键词
    text=self.wechat.keyword_reply[get_md5_value(content)]
elif content.startswith('msgs/'):
    text='天哪，我还不能处理非文本消息~'
else: #不存在调用图灵机器人 API
    text=self.call_tuling(content, get_md5_value(user['NickName']))
data={
```

```
        "msgType": 1,
        "data": text
    }
    t=random.uniform(1, 4)                  #随机休眠一定的时间再回复
    time.sleep(t)
    self.send msg(data, user['UserName'])  #发送消息
        self.Log.info('to:%s,send:%s' % (user['NickName'], text)) #记录发送信息
```

❹ 发送文本消息函数

```
    def webwxsendmsg(self, word, to='filehelper'):
        """
        发送消息
        :param word: 消息内容
        :param to: 接收人 ID
        :return: Dict or None
        """
        dic=None
        flag=0
        while flag < 2:  #出错会进行一次重发
            try:
                url=self.wx conf['API webwxsendmsg'] + \
                     '?pass ticket=%s' % (self.pass ticket)
                clientMsgId=str(int(time.time() * 1000)) + \
                            str(random.random())[:5].replace('.', '')
                params={
                    'BaseRequest': self.get base request(),
                    'Msg': {
                        "Type": 1,
                        "Content": word,
                        "FromUserName": self.User['UserName'],
                        "ToUserName": to,
                        "LocalID": clientMsgId,
                        "ClientMsgId": clientMsgId
                    },
                    'Scene': 0
                }
                headers={'content-type': 'application/json; charset=UTF-8'}
                return post(url, params, True, headers)
            except:
                if dic:
                    self.Log.info(dic)
                flag=flag + 1
                self.Log.error(traceback.format exc())
         return None
```

其实现效果如图 8-5 所示。



图 8-5　自动回复演示

8.4.2　群发消息、定时发送消息、好友状态检测

对于个人微信用户而言，在节日时经常会群发消息，可是千篇一律的祝福不仅无法体现出心意，反而会让接收者厌烦，但一个一个发送又太过烦琐，本程序实现的群发消息可根据接收者昵称或备注自动填充，做到发送的每一条消息都不一样，接收者也更容易接收。

定时发送消息的需求主要为自动报时，防止因长时间没有消息被微信认为挂机而强制下线等，一般可通过此方法延长机器人的运行时长。

在群发消息时，若遇到好友状态异常，例如拉黑或删除等，微信会发送系统消息，提示该好友状态异常，此时可通过检测微信返回消息，并给异常好友不同的备注来实现好友状态检测功能。

注意：在群发消息时需要注意发送频率，编者经过几次测试，发现有 10 个好友，在 1～4 秒时间段随机取一个时间进行休眠，连续 12 小时发送文本消息未出现因发送频繁被限制发送的情况，因此本程序限制随机时间为 1～4 秒。文件消息需延长此时间，可以通过配置文件修改文本消息休眠时长。

❶ 发送消息线程

```
def process(self):
    text="Never lie to someone who trust you. Never trust someone who lies
    to you.  to:%s"
    for contact in self.wechat.ContactList:
        #self.wechat.send_text(contact['UserName'], text % contact['NickName'])
```

```
            self.wechat.Log.info("send msg to:%s,text:%s" % (contact['NickName'],
            text % contact['NickName']))
            t=random.uniform(1, 4)
            time.sleep(t)     #随机休眠一段时间，避免微信认为操作违法
        #每小时向文件助手发送一个消息，避免长时间无操作被微信认为挂机，从而强制下线
        current minutes=int(get now time('%M'))
        if current minutes==59:
            self.wechat.send text('filehelper', '主人主人,为您报时,当前时间:%s' %
            get now time())
        time.sleep(60)
```

注意：在好友多的情况下，群发消息因为需要休眠比较耗时，一般采用多线程实现群发，避免影响主线程同步消息断开。在实际开发中，发送消息任务一般通过 redis 等消息队列通知发送消息内容，机器人获取到之后进行发送。

❷ 发送结果

群发消息演示如图 8-6 所示。

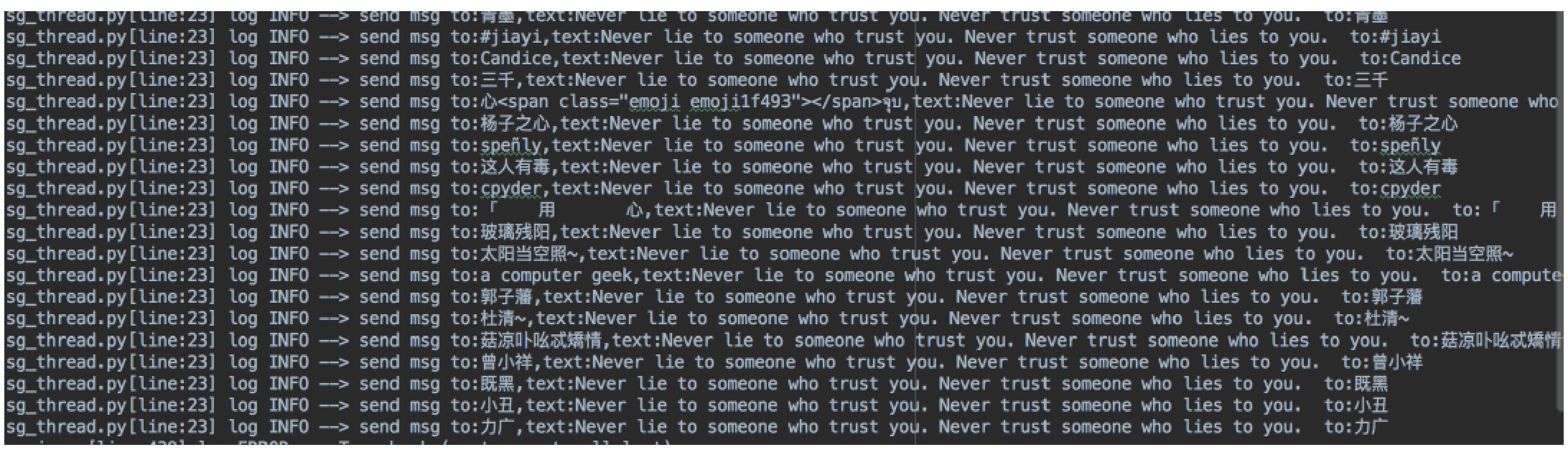


图 8-6　群发消息演示

注意：为避免给好友带来骚扰，上述代码中调用发送函数的语句已注释，通过 Log 打印要发送的消息展示发送结果。

❸ 好友状态检测

```
    SYS BLACK LIST CONTACT='消息已发出，但被对方拒收了。'
    SYS_DELETE_CONTACT='开启了朋友验证，你还不是他（她）朋友。请先发送朋友验证请求，
    对方验证通过后才能聊天'
    if msg['MsgType']==self.wechat.wx conf['MSGTYPE SYS']:
            if content==Constant.SYS_BLACK_LIST_CONTACT:    #黑名单
                res=self.wechat.webwxoplog(msg['FromUserName'],
                remark_name= 'A-拉黑-%s' % user['NickName'])
                self.wechat.Log.info('[黑名单]给%s 设置备注:%s'%(user['NickName'], res))
            elif Constant.SYS DELETE CONTACT in content:    #被删除好友
                res=self.wechat.webwxoplog(msg['FromUserName'],
                remark name= 'A-删除-%s' % user['NickName'])
                self.Log.info('[删除好友]给%s 设置备注:%s' % (user['NickName'], res))
            elif content.startswith(Constant.SYS ACCESS VERIFY INFO START):
                                                          #新添加好友
```

```
            data={
                "msgType": 1,
                "data": '%s 你好，终于等到你~' % user['NickName']
            }
            self.send_msg(data, user['UserName'])
            self.Log.info('[新增好友]给%s 发送欢迎语' % (user['NickName']))
```

8.4.3　自动邀请好友加入群聊

对于一些公司或个人的特殊需求，需要根据某些属性自动邀请用户进入不同的群聊，例如华中区、华南区等，此时需要机器人根据用户发送的特殊指令+关键词（例如我要入群+华中）自动查询所属的群聊，然后给该用户发送入群链接，并引导用户单击加入。

程序设计包括好友消息分析、关键词和群聊对应关系、自动发送入群邀请、自动发送提示语句等，具体如下：

❶ 初始化关键词和群聊对应关系

```
self.enter_group_keyword={  #自动入群关键词和群聊对应字典
            '1 群': '测试 1 群',
            '2 群': '测试 2 群',
        }
```

❷ 根据群名查找群聊

```
def get_group_by_name(self, name):
       for member in self.GroupList:
           if member['NickName']==name:
               return member
       return None
}
```

❸ 更新群聊 API 函数

```
   def webwxupdatechatroom(self, room_user_name, add_arr="", del_arr="",
   invite_arr="", topic=None):
        flag=0
        dic=None
        while flag < 2:
            try:
                params={
                    'BaseRequest': self.get_base_request(),
                    'ChatRoomName': room_user_name
                }
                base_url=self.wx_conf['API_webwxupdatechatroom'] + '?fun=%s'
                if invite_arr:  #发送入群邀请链接
                    url=base_url % 'invitemember'
                    params['InviteMemberList']=invite_arr
```

```
                elif add_arr:  #直接添加进群
                    url=base_url % 'addmember'
                    params['AddMemberList']=add_arr
                elif del_arr:  #删除群成员
                    url=base_url % 'delmember'
                    params['DelMemberList']=add_arr
                elif topic:  #修改群名
                    url=base_url % 'modtopic'
                    params['NewTopic']=topic
                headers={'content-type': 'application/json; charset=UTF-8'}
                url += '&lang=zh_CN&pass_ticket=' + self.pass_ticket
                dic=post(url, params, True, headers)
                #Ret 0        成功
                #    -1|-2    失败  群开启群主验证
                #    1205     失败  操作频繁
                if dic['BaseResponse']['Ret']!= 0:
                    group=self.get_group_by_id(room_user_name)
                    self.Log.error('[%s]更新群聊失败,错误代码:%s,错误原因:%s' %
                   (group['NickName'], dic['BaseResponse']['Ret'],
                   dic['BaseResponse']['ErrMsg']))
                    self.Log.info(str(dic))
                return dic['BaseResponse']['Ret']==0
            except:
                flag=flag+1
                if dic:
                    self.Log.info(dic)
                self.Log.error(traceback.format_exc())
        return False
}
```

❹ 好友消息分析

```
#Constant.STRING_ENTER_GROUP_KEYWORD='我要入群+'
if Constant.STRING_ENTER_GROUP_KEYWORD in content:     #关键词入群
    keyword=content.split(Constant.STRING_ENTER_GROUP_KEYWORD)[1]
    #在关键词列表中查找所属群
    room_name=self.wechat.enter_group_keyword[keyword] if keyword in self
    .wechat.enter_group_keyword else None
    remind_msg=Constant.STRING_ENTER_GROUP_NOT_FOUND    #换个关键词试试
    if room_name:
        group=self.wechat.get_group_by_name(room_name)
        if group:
            res=self.wechat.webwxupdatechatroom(group['UserName'], invite_arr
           =user['UserName'])
            if res:
                remind_msg=Constant.STRING_ENTER_GROUP_SUCCESS#邀请链接进群
```

```
            else:
                remind_msg=Constant.STRING_ENTER_GROUP_UNKNOWN_ERROR#出错
    data={
        "msgType": 1,
        "data": remind_msg
    }
    t=random.uniform(1, 4)
    time.sleep(t)
    self.send msg(data, user['UserName'])
    self.Log.info('to:%s,send:%s' % (user['NickName'], remind_msg))}
```

其实现效果如图 8-7 所示。



图 8-7　自动发送入群邀请

第9章

图像处理——生成二维码和验证码

视频讲解

9.1 二维码介绍

二维码（二维条码）是指在一维条码的基础上扩展出的另一维具有可读性的条码，使用黑白矩形图案表示二进制数据，被设备扫描后可获取其中所包含的信息。一维条码的宽度记载着数据，而其长度没有记载数据。二维条码的长度、宽度均记载着数据。二维条码有一维条码没有的“定位点”和“容错机制”。容错机制使得在即使没有辨识到全部的条码或者条码有污损时也可以正确地还原条码上的信息。二维码的种类很多，不同的机构开发出的二维码具有不同的结构以及编写、读取方法。

❶ 堆叠式二维码

堆叠式二维码例如 PDF417（证件及卡片等大容量、高可靠性信息自动存储、携带并可用机器自动识别）、Code49、Code16K、Ultracode 等。

❷ 矩阵式二维码

矩阵式二维码例如 QR 码、Code One、Aztec、Data Matrix 等。

本章使用的二维码是 QR Code（Quick Response Code），学名为快速响应矩阵码，它是二维条码的一种。QR 二维码目前在很多地方有着广泛的应用，例如通过微信二维码加好友、将应用软件的下载地址做成二维码，等等。QR 二维码是于 1994 年由日本的 Denso-Wave 公司发明的。QR 是英文 Quick Response 的缩写，即快速响应的意思，源自发明者希望 QR 码可让其内容快速被解码。QR 二维码比普通条码可储存更多资料，也无须像普通条码那样在扫描时需要直线对准扫描器。

QR 码呈正方形，只有黑、白两色，在 3 个角落印有较小、像“回”字的正方形图案。这 3 个图案是帮助解码软件定位的图案，用户不需要对准，无论以何角度扫描，数据仍可被正确读取。

QR 二维码的结构如下。

（1）版本信息：version1（21×21），version2，…，version40，一共 40 个版本。版本代表每行有多少码元模块，每一个版本比前一个版本增加 4 个码元模块，计算公式为(n−1)×4+21，每个码元模块存储一个二进制 0 或者 1，黑色模块表示二进制“1”，白色模块表示二进制“0”，例如 version1 表示每一行有 21 个码元模块。

（2）格式信息：存储容错级别有 L（7%）、M（15%）、Q（25%）、R（35%）。容错指允许存储的二维码信息出现重复部分，级别越高，重复信息所占的比例越高。其目的是即使二维码被图标遮住一部分，一样可以获取全部二维码内容。有图片的二维码，图片不算二维码的一部分，它遮住一部分码元，但还是可以扫描到所有内容。

（3）数据和纠错码字：实际保存的二维码信息和纠错码字（用于修正二维码损坏带来的错误，也就是说当码元被图片遮住时可以通过纠错码字来找回）。

（4）位置探测图形：用于对二维码的定位。位置探测图形用于标记矩形大小，3 个图形确定一个矩形。

上述二维码结构信息按照一定的编码规则变成二进制，通过黑、白色形成矩形。

除了标准的 QR 码之外，还存在一种称为“微型 QR 码”的格式，它是 QR 码标准的缩小版本，主要是为了无法处理较大型扫描的应用而设计。微型 QR 码同样有多种标准，最多可存储 35 个字符。

9.2　二维码生成和解析关键技术

9.2.1　qrcode 库的使用

❶ 安装 qrcode 库

如果要用 Python 生成二维码，首先需要下载 Python 的二维码库 qrcode。qrcode 库是用于生成二维码图像的 Python 第三方库。

qrcode 二维码生成包的安装如下（在命令行 cmd 中）：

```
C:\> pip install qrcode
```

查看 qrcode 库的安装信息，如图 9-1 所示。

```
C:\> pip show qrcode
```

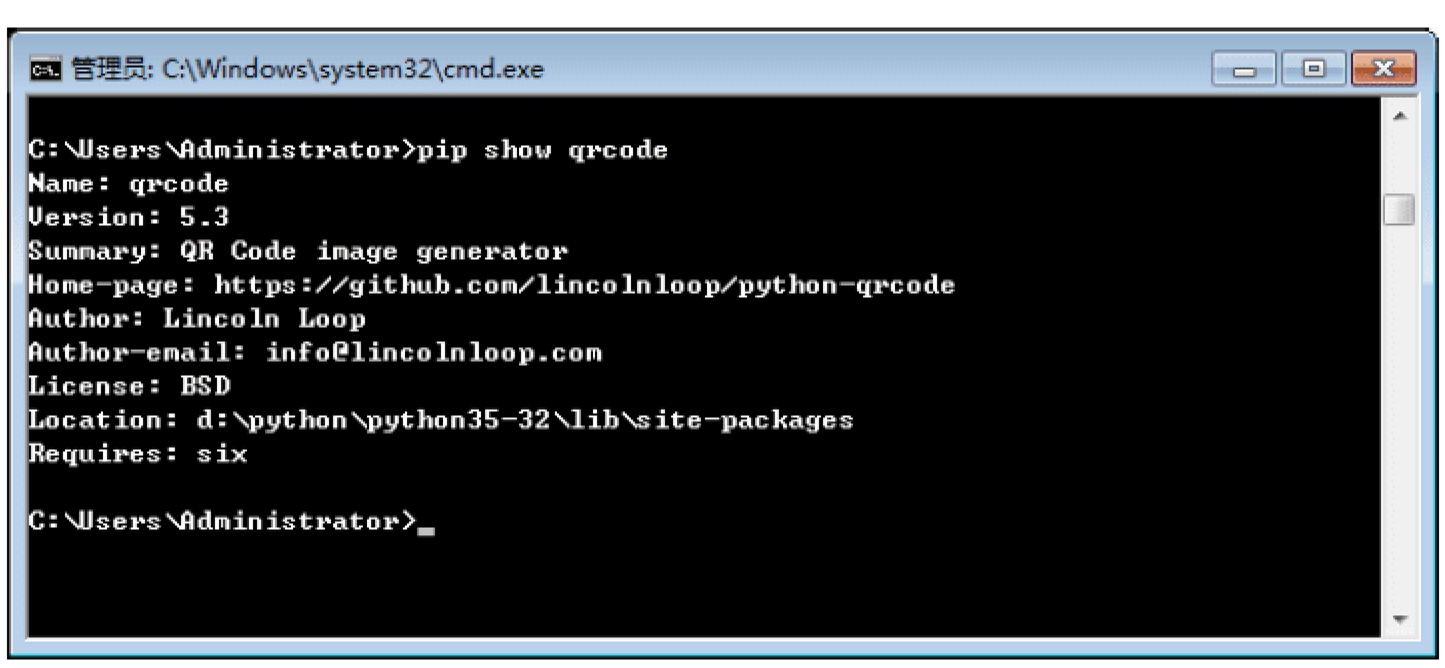


图 9-1　查看 qrcode 库的安装信息

❷ 生成二维码

导入 qrcode 模块后，make()函数返回一个 qrcode.image.pil.PilImage 对象，调用 make()函数生成一个二维码图片对象，如图 9-2 所示，最后调用图片对象的 save()函数就可以将生成的二维码保存下来。代码如下：

```
import qrcode
img=qrcode.make("http://www.zut.edu.cn")
img.save("xinxing.png")
```

图 9-2　生成的二维码图片 xinxing.png

上面是按照 qrcode 默认的方式生成二维码，如果希望生成不同尺寸的二维码，则需要使用 QRCode 类，如下：

```
QRCode(version=None, error correction=constants.ERROR CORRECT M, box size=10,
border=4, image_factory=None)
```

version 表示二维码的版本号，二维码总共有 1～40 个版本，最小的版本号是 1，对应的尺寸是 21×21，每增加一个版本会增加 4 个尺寸。这里所说的尺寸不是生成图片的大小，而是指二维码的长宽被平均分为多少份。

error_correction 指的是纠错容量，这就是为什么二维码上面放一个小图标也能扫出来的原因，纠错容量有 4 个级别，分别如下。

- ERROR_CORRECT_L L 级别：7%或更少的错误能修正。
- ERROR_CORRECT_M M 级别：15%或更少的错误能修正，也是 qrcode 的默认级别。
- ERROR_CORRECT_Q Q 级别：25%或更少的错误能修正。
- ERROR_CORRECT_H H 级别：30%或更少的错误能修正。

box_size 指的是生成图片的像素。

border 表示二维码的边框宽度，4 是最小值。

image_factory 参数是一个继承于 qrcode.image.base.BaseImage 的类，用于控制 make_image()函数返回的图像实例。image_factory 参数可以选择的类保存在模块根目录的 image 文件夹下。image 文件夹里面有 5 个.py 文件，其中一个为__init__.py，一个为 base.py，还有 pil.py（提供了默认的 qrcode.image.pil.PilImage 类）、pure.py（提供了 qrcode.image.pure.PymagingImage 类）、svg.py（提供了 SvgFragmentImage、SvgImage 和 SvgPathImage

几个类）。

注：实际上，make()函数也是通过实例化一个 QRCode 对象来生成二维码的。在调用 make()的时候也可以传入初始化参数。

例子如下：

```
import qrcode
qr=qrcode.QRCode(
    version=1,
    error_correction=qrcode.constants.ERROR_CORRECT_L,
    box_size=10,
    border=4,
)
qr.add_data('http://www.zut.edu.cn')
qr.make(fit=True)
img=qr.make_image()
img.save('xinxingzhao.png')
```

说明：QRCode 对象的 make_image()函数可以通过改变 fill_color 和 back_color 参数来改变所生成图片的背景颜色和格子颜色。

❸ 生成其他类型的二维码

用户可以将二维码图片转化为 SVG（矢量图）。qrcode 可以生成 3 种不同的 SVG 图像，一种是用路径表示的 SVG，一种是用矩形集合表示的完整 SVG 文件，还有一种是用矩形集合表示的 SVG 片段。第 1 种用路径表示的 SVG 其实就是矢量图，可以在图像放大的时候保持图片质量，而另外两种可能会在格子之间出现空隙。

这 3 种分别对应了 svg.py 中的 SvgPathImage、SvgImage 和 SvgFragmentImage 类。在调用 qrcode.make()函数或者实例化 QRCode 时当作参数传入就可以了。

另外还有 qrcode.image.svg.SvgFillImage 和 qrcode.img.svg.SvgPathFillImage，分别继承自 SvgImage 和 SvgPathImage。这两个并没有其他改变，只不过是默认把背景颜色设置为白色而已。

```
import qrcode
import qrcode.image.svg
if method=='basic':
    #简单的工厂，只有一套
    factory=qrcode.image.svg.SvgImage
elif method=='fragment':
    #碎片工厂（也只是一组矩形）
    factory=qrcode.image.svg.SvgFragmentImage
else:
    #组合路径工厂，修复缩放时可能出现的空白
    factory=qrcode.image.svg.SvgPathImage
    img=qrcode.make('xinxingzhao', image_factory=factory)
```

9.2.2 PIL 库的使用

生成二维码图像同时依赖于 PIL 库，PIL（Python Imaging Library，图像处理类库）提供了通用的图像处理功能，以及大量有用的基本图像操作，例如图像的缩放、裁剪、旋转和颜色转换等。PIL 是 Python 语言的第三方库，安装 PIL 库的方法如下，需要安装的库的名字是 pillow。

```
C:\> pip install pillow 或者 pip3 install pillow
```

PIL 库支持图像的存储、显示和处理，能够处理几乎所有图片格式，可以完成对图像的缩放、剪裁、叠加以及向图像添加线条和文字等操作。

PIL 库主要实现图像归档和图像处理两方面的功能需求。

- 图像归档：对图像进行批处理、生成图像预览、转换图像格式等。
- 图像处理：图像的基本处理、像素处理、颜色处理等。

根据不同功能，PIL 库共包括 21 个与图像相关的类，这些类可以被看作是子库或 PIL 库中的模块，各模块如下：

Image、ImageChops、ImageCrackCode、ImageDraw、ImageEnhance、ImageFile、ImageFileIO、ImageFilter、ImageFont、ImageGrab、ImageOps、ImagePath、ImageSequence、ImageStat、ImageTk、ImageWin、PSDraw

下面介绍几种最常用的模块。

❶ Image 模块

Image 模块是 PIL 中最重要的模块，它提供了诸多图像操作功能，例如创建、打开、显示、保存图像等功能，合成、裁剪、滤波等功能，获取图像属性等功能。

PIL 中的 Image 模块提供了 Image 类，用户可以使用 Image 类从大多数图像格式的文件中读取数据，然后写到最常见的图像格式文件中。如果要读取一幅图像，可以使用以下代码：

```
from PIL import Image
pil_im=Image.open('empire.jpg')
```

上述代码的返回值 pil_im 是一个 PIL 图像对象。

用户也可以直接使用 Image.new(mode,size,color=None)创建图像对象，color 的默认值是黑色。

```
newIm=Image.new('RGB', (640, 480), (255, 0, 0))    #新建一个 Image 对象
```

这里新建了一个红色背景、大小为（640, 480）的 RGB 空白图像。

图像的颜色转换可以使用 Image 类的 convert()方法来实现。如果要读取一幅图像，并将其转换成灰度图像，只需要加上 convert('L')即可，例如：

```
pil_im=Image.open('empire.jpg').convert('L')        #转换成灰度图像
```

❷ ImageChops 模块

ImageChops 模块包含一些算术图形操作，即 channel operations("chops")。这些操作可

用于诸多目的，例如图像特效、图像组合、算法绘图等。通道操作只用于位图（比如 L 模式和 RGB 模式）。大多数通道操作有一个或者两个图像参数，返回一个新的图像。

每张图片都是由一个或者多个数据通道构成的。以 RGB 图像为例，每张图片都是由 3 个数据通道构成的，分别为 R、G 和 B 通道。对于灰度图像，只有一个通道。

ImageChops 模块的使用如下：

```
from PIL import Image
im=Image.open('D:\\1.jpg')
from PIL import ImageChops
im_dup=ImageChops.duplicate(im)          #复制图像，返回给定图像的副本
print(im_dup.mode)                       #输出模式：'RGB'
im_diff=ImageChops.difference(im,im_dup) #返回两幅图像之间像素差的绝对值形成的图像
im_diff.show()
```

由于图像 im_dup 是由 im 复制过来的，所以它们的差为 0，图像 im_diff 在显示时为黑图。

❸ ImageDraw 模块

ImageDraw 模块为 Image 对象提供了基本的图形处理功能，例如它可以为图像添加几何图形。

ImageDraw 模块的使用如下：

```
from PIL import Image, ImageDraw
im=Image.open('D:\\1.jpg')
draw=ImageDraw.Draw(im)
draw.line((0,0)+im.size, fill=128)
draw.line((0, im.size[1], im.size[0], 0), fill=128)
im.show()
```

其结果是在原有图像上画了两条对角线。

❹ ImageEnhance 模块

ImageEnhance 模块包括了一些用于图像增强的类，它们分别为 Color 类、Brightness 类、Contrast 类和 Sharpness 类。

ImageEnhance 模块的使用如下：

```
from PIL import Image, ImageEnhance
im=Image.open('D:\\1.jpg')
enhancer=ImageEnhance.Brightness(im)
im0=enhancer.enhance(0.5)
im0.show()
```

其结果是图像 im0 的亮度为图像 im 的一半。

❺ ImageFile 模块

ImageFile 模块为图像的打开和保存提供了相关支持功能。

❻ ImageFilter 模块

ImageFilter 模块包括各种滤波器的预定义集合，与 Image 类的 filter()方法一起使用。

该模块包含一些图像增强的滤波器，例如 BLUR、CONTOUR、DETAIL、EDGE_ENHANCE、EDGE_ENHANCE_MORE、EMBOSS、FIND_EDGES、SMOOTH、SMOOTH_MORE 和 SHARPEN。

ImageFilter 模块的使用如下：

```
from PIL import Image
im=Image.open('D:\\1.jpg')
from PIL import ImageFilter
imout=im.filter(ImageFilter.BLUR)
print(imout.size)#图像的尺寸大小为(300, 450)，它是一个二元组，即水平和垂直方向上的像素数
imout.show()
```

❼ ImageFont 模块

ImageFont 模块定义了一个同名的类，即 ImageFont 类。在这个类的实例中存储着 bitmap 字体，需要和 ImageDraw 类的 text()方法一起使用。

在 Python 中通过使用 PIL 库中的这些模块和类来处理和使用图像。

9.3 二维码生成和解析程序设计的步骤

9.3.1 生成带有图标的二维码

事先准备一个 logo 图标，使用下面的程序生成带有 logo 图标的二维码。

```
import qrcode
from PIL import Image
import os, sys
def gen qrcode(string, path, logo=""):
    """
    生成中间带 logo 的二维码
    需要安装 qrcode、PIL 库
    @参数 string: 二维码字符串
    @参数 path: 生成的二维码保存路径
    @参数 logo: logo 文件路径
    @return: None
    """
    #初步生成二维码图像
    qr=qrcode.QRCode(
        version=2,
        error correction=qrcode.constants.ERROR CORRECT H,
        box size=8,
        border=1
    )
    qr.add_data(string)
```

```
        qr.make(fit=True)
        #获得 Image 实例并把颜色模式转换为 RGBA
        img=qr.make image()
        img=img.convert("RGBA")
        if logo and os.path.exists(logo):
            try:
                icon=Image.open(logo)          #打开填充的 logo 文件
                img w, img h=img.size
            except Exception as e:
                print(e)
                sys.exit(1)
            factor=4
            #计算 logo 的尺寸
            size w=int(img w/factor)
            size h=int(img h/factor)
            #比较并重新设置 logo 文件的尺寸
            icon w,icon h=icon.size
            if icon w>size w:
                icon w=size w
            if icon h>size h:
                icon h=size h
            icon=icon.resize((icon w,icon h), Image.ANTIALIAS)
            #计算 logo 的位置，并复制到二维码图像中
            w=int((img w-icon w)/2)
            h=int((img h-icon h)/2)
            icon=icon.convert("RGBA")
            img.paste(icon, (w, h),icon)
            #保存二维码
            img.save(path)                   #例如 qrcode.png
    if __name__=="__main__":
        info=" http://www.zut.edu.cn "
        pic_path="qrcode.png"                #生成的带有图标的二维码图片，如图 9-3 所示
        logo_path="logo.png"                 #用于填充的图标
        gen_qrcode(info, pic_path,logo_path)
```

图 9-3　生成的带有图标的二维码图片 qrcode.png

9.3.2 Python 解析二维码图片

解析二维码图片的信息需要使用 zbarlight（二维码解析包），zbarlight 的安装如下：

```
pip install zbarlight
```

注意，zbarlight 二维码解析包仅仅支持 Python2.7 以下的版本。

```
import zbar
def decode qrcode(path):
    """
    解析二维码信息
    @参数 path: 二维码图片路径
    @return: 二维码信息
    """
    scanner=zbar.ImageScanner()          #创建图片扫描对象
    scanner.parse config('enable')       #设置对象属性
    img=Image.open(path).convert('L')    #打开含有二维码的图片
    width, height=img.size               #获取图片的尺寸
    #建立 zbar 图片对象并扫描转换为字节信息
    qrCode=zbar.Image(width, height, 'Y800', img.tobytes())
    scanner.scan(qrCode)
    #组装解码信息
    data=''
    for s in qrCode:
        data += s.data
    del img                              #删除图片对象
    return data                          #输出解码结果
if __name__=="__main__":
    info=" http://www.zut.edu.cn "
    pic_path="qrcode.png"                #生成的带有图标的二维码图片
    logo path="logo.png"                 #用于填充的图标
    gen qrcode(info, pic path,logo path)
    print(decode qrcode(pic path))
                        #得到二维码内的文本信息，即网址“http://www.zut.edu.cn”
```

9.4 用 Python 生成验证码图片

基本上，大家使用每一种网络服务都会遇到验证码，一般是网站为了防止恶意注册、发帖而设置的验证手段。其生成原理是将一串随机产生的数字或符号生成一幅图片，图片里加上一些干扰像素（防止 OCR）。下面详细讲解如何生成验证码。

视频讲解

通常，除了配置好的 Python 环境以外，还需要配有 Python 中的 PIL 库，这是 Python 中专门用来处理图片的库。

如果要生成验证码图片，首先要生成一个随机字符串，包含 26 个字母和 10 个数字。

```
#用来随机生成一个字符串
def gene text():
    #source=list(string.letters)
    #source =[ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l',
    'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
    source=list(string.ascii letters)
    for index in range(0,10):
        source.append(str(index))
    return ''.join(random.sample(source,number))  #number 是生成验证码的位数
```

然后要创建一个图片，写入字符串，需要注意这里面的字体是不同系统而定的，如果没有找到系统字体路径，也可以不设置。接下来要在图片上画几条干扰线。

最后创建扭曲，加上滤镜，用来增强验证码的效果。下面是用程序生成的一个验证码。



完整的代码如下：

```
#coding=utf-8
import random, string, sys, math
from PIL import Image,ImageDraw,ImageFont,ImageFilter
font_path=' C:\Windows\Fonts\simfang.ttf ' #字体的位置
number=4                                   #生成几位数的验证码
size=(80,30)                               #生成验证码图片的高度和宽度
bgcolor=(255,255,255)                      #背景颜色，默认为白色
fontcolor=(0,0,255)                        #字体颜色，默认为蓝色
linecolor=(255,0,0)                        #干扰线颜色，默认为红色
draw line=True                             #是否要加入干扰线
line number=(1,5)                          #加入干扰线条数的上/下限
#用来随机生成一个字符串
def gene text():
    #source=list(string.letters)
    #source =[ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l',
    'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
    source=list(string.ascii letters)
    for index in range(0,10):
        source.append(str(index))
    return ''.join(random.sample(source,number))#number 是生成验证码的位数
#用来绘制干扰线
def gene line(draw,width,height):
    begin=(random.randint(0, width), random.randint(0, height))
    end=(random.randint(0, width), random.randint(0, height))
    draw.line([begin, end], fill=linecolor)
#生成验证码
def gene_code():
```

```
    width,height=size                                        #宽和高
    image=Image.new('RGBA',(width,height),bgcolor)           #创建图片
    font=ImageFont.truetype(font_path,25)                    #验证码的字体
    draw=ImageDraw.Draw(image)                               #创建画笔
    text=gene_text()                                         #生成字符串
    font width, font height=font.getsize(text)
    draw.text(((width - font width)/number, (height - font height) / number),text,
            font=font,fill=fontcolor)                        #填充字符串
    if draw line:
        gene line(draw,width,height)
    image=image.transform((width+20,height+10), Image.AFFINE,
         (1,-0.3,0,-0.1,1,0),Image.BILINEAR)                 #创建扭曲
    image=image.filter(ImageFilter.EDGE_ENHANCE_MORE)        #滤镜，边界加强
    image.save('idencode.png')                               #保存验证码图片
if __name__=="__main__":
    gene_code()
```

通过上面的两个例子可见 Python 的图像处理功能十分完善。

第10章

益智游戏——连连看游戏

10.1 连连看游戏介绍

视频讲解

“连连看”是源自中国台湾的桌面小游戏，自从流入中国大陆以来风靡一时，也吸引了众多程序员开发出多种版本的“连连看”。“连连看”考验的是玩家的眼力，在有限的时间内只要把所有能连接的相同图案两个一对地找出来，每找出一对，它们就会自动消失，把所有的图案全部消完即可获得胜利。所谓能够连接，指的是无论横向或者纵向，从一个图案到另一个图案之间的连线不能超过两个弯，其中，连线不能从尚未消去的图案上经过。

连连看游戏的规则如下：

- 两个选中的方块是相同的。
- 两个选中的方块之间连接线的折点不超过两个（连接线由 X 轴和 Y 轴的平行线组成）。

本章开发连连看游戏，游戏效果如图 10-1 所示。

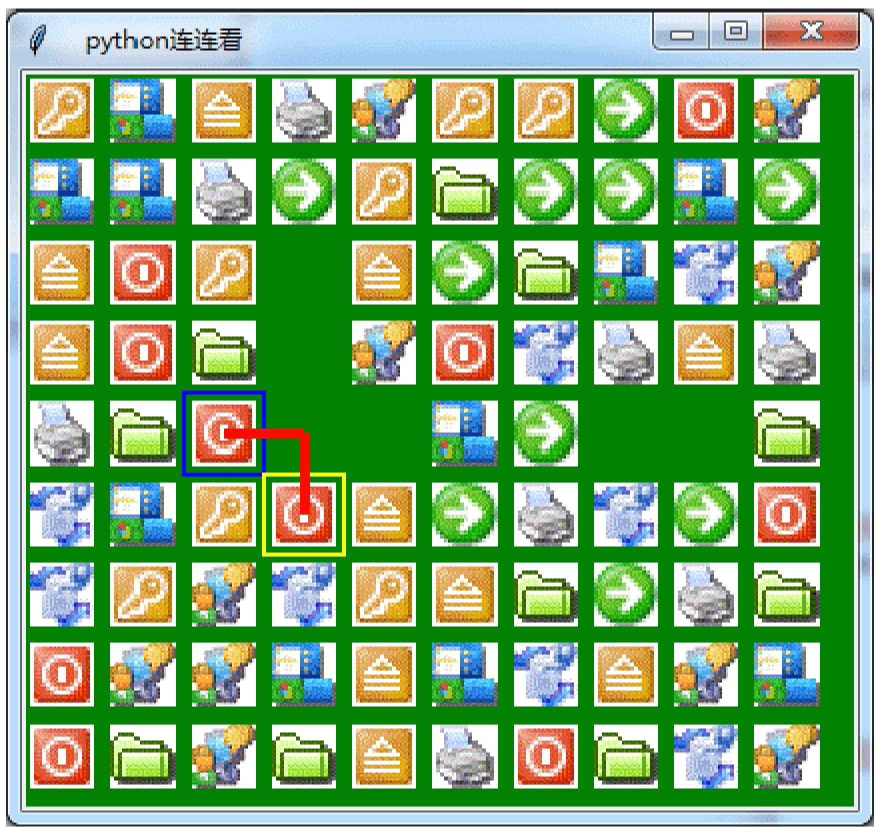


图 10-1　连连看游戏的运行界面

本游戏增加了智能查找功能，当玩家自己无法找到可以消去的两个方块时右击画面，系统则会提示可以消去的两个方块（被加上红色的边框线）。

10.2 程序设计的思路

❶ 图标方块的布局

游戏中有 10 种方块，如图 10-2 所示，而且每种方块有 10 个，可以先按顺序把每种图标方块（数字编号）排好放入列表 tmpMap（临时的地图）中，然后用 random.shuffle()打乱列表元素的顺序，依次从 tmpMap（临时的地图）中取一个图标方块放入地图 map 中。实际上，在程序内部是不需要认识图标方块的图像的，只需要用一个 ID 来表示，在运行界面上的图标图形是根据地图中的 ID 取资源里的图片画的。如果 ID 的值为空（" "），则说明此处已经被消除掉了。

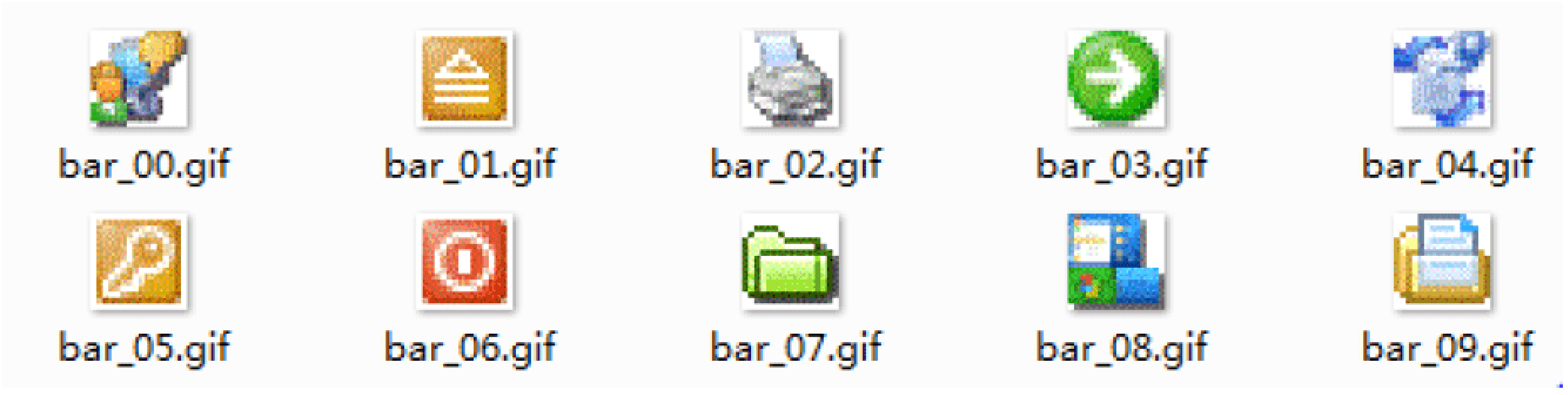


图 10-2　图标方块

```
imgs=[PhotoImage(file='H:\\连连看\\gif\\bar 0'+str(i)+'.gif') for i in range(0,10)]    #所有图标图案
```

所有图标图案存储在列表 imgs 中，地图 map 中是图标图案存储在列表 imgs 中的索引号。如果是 bar_02.gif 图标，在地图 map 中实际存储的是 2；如果是 bar_08.gif 图标，在地图 map 中实际存储的是 8。

```
#初始化地图，将地图中的所有方块区域位置置为空方块状态
map=[[" " for y in range(Height)]for x in range(Width)]
#存储图像对象
image map=[[" " for y in range(Height)]for x in range(Width)]
cv=Canvas(root, bg='green', width=610, height=610)
def create_map():#产生 map 地图
    global map
    #生成随机地图
    #将所有匹配成对的图标索引号放进一个临时的地图中
    tmpMap=[]
    m=(Width)*(Height)//10
    print('m=',m)
    for x in range(0,m):
```

```
        for i in range(0,10):            #每种方块有 10 个
          tmpMap.append(x)
    random.shuffle(tmpMap)               #生成随机地图
    for x in range(0,Width):
        for y in range(0,Height):
            map[x][y]=tmpMap[x*Height+y]  #从上面的临时地图中获取
```

❷ 连通算法

那么分析一下，可以看到连接一般有 3 种情况，如图 10-3 所示。

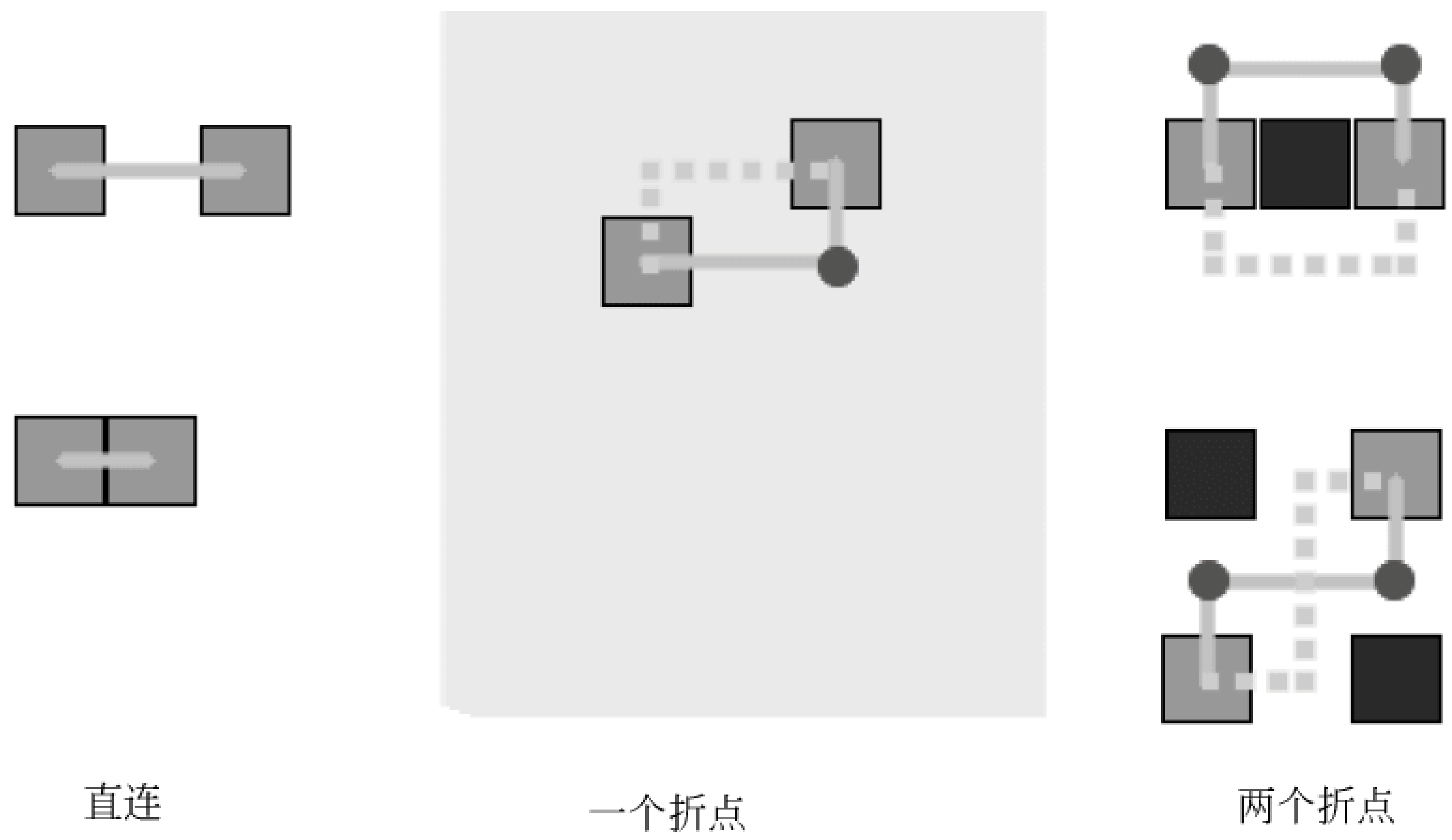


图 10-3　两个选中方块之间连接线的示意图

1）直连方式

在直连方式中，要求两个选中的方块 x 和 y 相同，即在一条直线上，并且之间没有其他任何图案的方块。直连方式在 3 种连接方式中最简单。

2）一个折点

其实，该情况相当于两个方块画出一个矩形，这两个方块是一对对角顶点，另外两个顶点中的某个顶点（即折点）如果可以同时和这两个方块直连，就说明可以“一折连通”。

3）两个折点

这种方式的两个折点（z1、z2）必定在两个目标点（两个选中的方块）p1、p2 所在的 x 方向或 y 方向的直线上。

按 p1(x1,y1)点向 4 个方向探测，例如向右探测，每次 x1+1，判断 z1(x1+1,y1)和 p2(x2,y2)点可否形成一个折点连通性，如果可以形成连通，则两个折点连通，否则直到超过图形右边界区域。假如超过图形右边界区域，则还需要判断两个折点在选中方块的右侧，且两个折点在图案区域之外连通情况是否存在。此时判断可以简化为判断 p2(x2,y2)是否可以水平直通到边界。

经过上面的分析，两个方块是否可以抵消的流程图如图 10-4 所示。

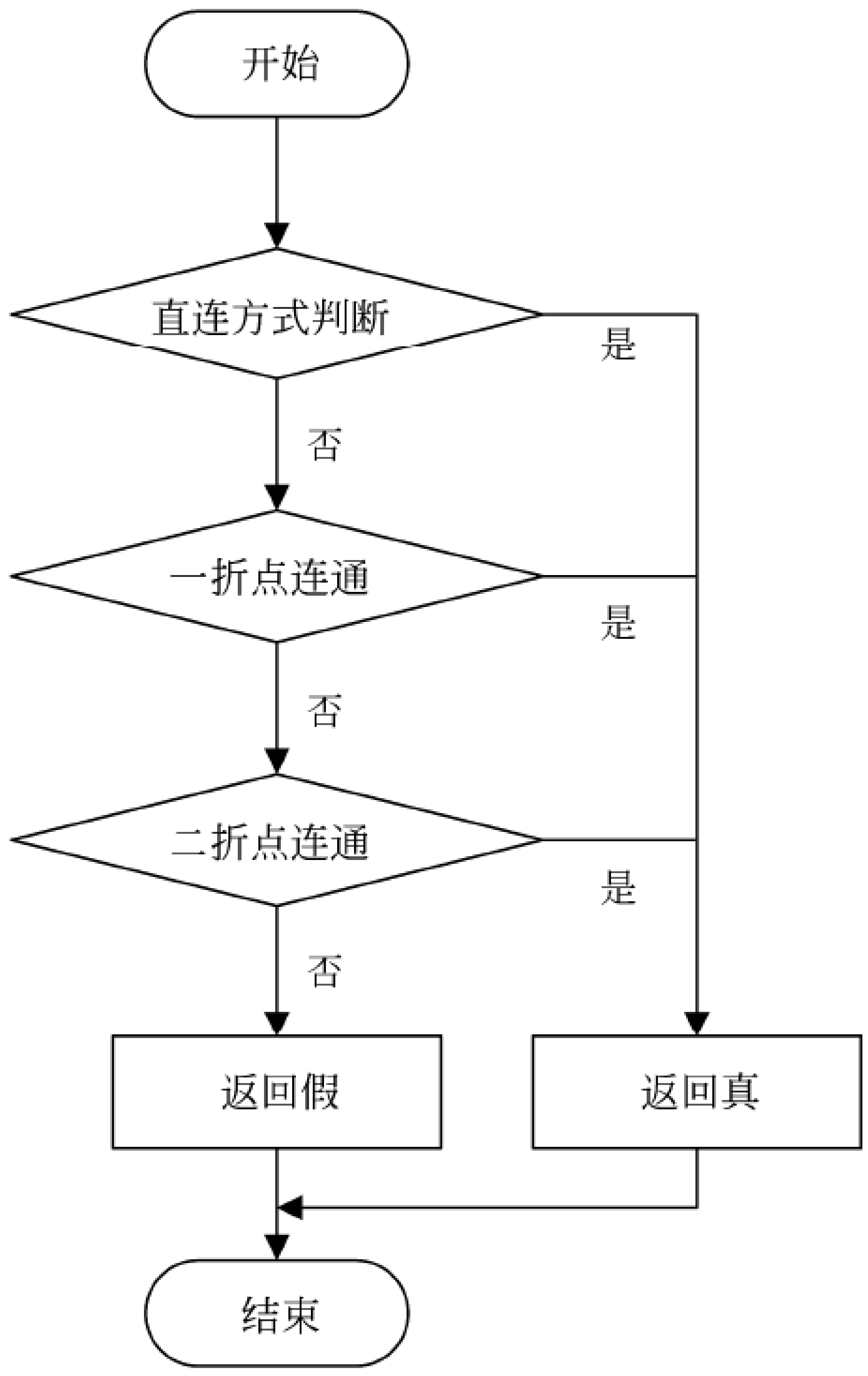


图 10-4　流程图

根据图 10-4 所示的流程图，对选中的两个方块（分别在(x1,y1)、(x2,y2)位置）是否可以抵消的判断如下实现。把该功能封装在 IsLink()方法里面，其代码如下：

```
'''
判断选中的两个方块是否可以消除
'''
def IsLink(p1,p2):
    if lineCheck(p1, p2):
        return True
    if secondLine(p1, p2):        #一个转弯（折点）的连通方式
        return True
    if triLine(p1, p2):           #两个转弯（折点）的连通方式
        return True
    return False
```

直连方式分为 x 或 y 相同的情况，同行同列情况消除的原理是如果两个相同的被消除方块之间的空格数 spaceCount 等于它们的(行/列差-1)，则两者可以连通。

```
class Point:
     #点类
     def __init__(self,x,y):
        self.x=x
        self.y=y
```

```
'''
* x 代表列，y 代表行
* param p1 第 1 个保存上次选中点坐标的点对象
* param p2 第 2 个保存上次选中点坐标的点对象
'''
#直接连通
def lineCheck(p1, p2):
    absDistance=0
    spaceCount=0
    if (p1.x==p2.x or p1.y==p2.y) :#同行同列吗
       print("同行同列的情况------")
       #同列的情况
       if (p1.x==p2.x and p1.y != p2.y) :
           print("同列的情况")
           #绝对距离（中间隔着的空格数）
           absDistance=abs(p1.y-p2.y)-1
           #正/负值
           if  p1.y-p2.y > 0 :
               zf=-1
           else:
               zf=1
           for i in range(1,absDistance+1):
               if (map[p1.x][p1.y + i * zf]==" "):
                   #空格数加 1
                   spaceCount+=1
               else:
                   break;#遇到阻碍就不用再探测了
       elif (p1.y==p2.y and p1.x!=p2.x):    #同行的情况
            print("同行的情况")
            absDistance=abs(p1.x-p2.x)-1
            #正/负值
            if  p1.x-p2.x > 0 :
               zf=-1
            else:
               zf=1
            for i in range(1,absDistance+1):
                if (map[p1.x + i * zf][p1.y]==" "):
                    #空格数加 1
                    spaceCount+=1
                else:
                    break;              #遇到阻碍就不用再探测了
       if (spaceCount==absDistance) :
           #可连通
           print(absDistance,spaceCount)
```

```
            print("行/列可直接连通")
            return True
        else:
            print("行/列不能消除！")
            return False
    else:
        #不是同行同列的情况，所以直接返回 False
        return False;
```

一个折点连通使用 OneCornerLink()实现判断。其实相当于两个方块画出一个矩形，这两个方块是一对对角顶点，见图 10-5 中两个黑色目标方块的连通情况，右上角打叉的位置就是折点，左下角打叉的位置不能与左上角的黑色目标方块连通，所以不能作为折点。

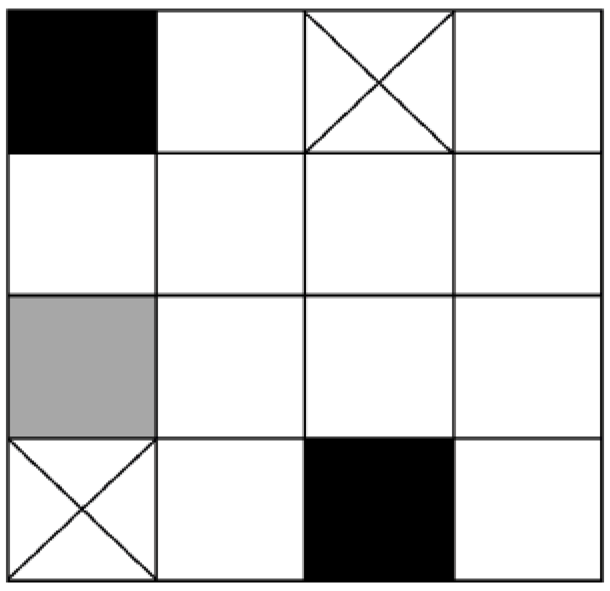


图 10-5　一个折点连通示意图

如果找到，则把折点放入 linePointStack 列表中。

```
#第 2 种，一个折点连通（直角连通）
'''
一个折点连通
@param first:选中的第 1 个点
@param second:选中的第 2 个点
'''
def secondLine(p1, p2):
    #第 1 个直角检查点
    checkP=Point(p1.x, p2.y)
    #第 2 个直角检查点
    checkP2=Point(p2.x, p1.y);
    #第 1 个直角点检测
    if (map[checkP.x][checkP.y]==" "):
        if (lineCheck(p1, checkP) and lineCheck(checkP, p2)):
            linePointStack.append(checkP)
            print("直角消除 OK",checkP.x,checkP.y)
            return True
    #第 2 个直角点检测
    if (map[checkP2.x][checkP2.y]==" "):
        if (lineCheck(p1, checkP2) and lineCheck(checkP2, p2)):
            linePointStack.append(checkP2)
```

```
            print("直角消除 OK",checkP2.x,checkP2.y)
            return True
    print("不能直角消除" )
    return False
```

两个折点连通（双直角连通）使用 TwoCornerLink()实现判断。双直角连通的判定可以分两步进行：

（1）在 p1 点周围的 4 个方向寻找空块 checkP 点。

（2）调用 OneCornerLink(checkP, p2) 检测 checkP 与 p2 点可否形成一个折点连通性。

两个折点连通即遍历 p1 点周围 4 个方向的空格，使之成为 checkP 点，然后调用 OneCornerLink(checkP, p2)判定是否为真，如果为真，可以双直角连通，若所有的空格都遍历完还没有找到，则失败。

如果找到，则把两个折点放入 linePointStack 列表中。

```
'''
#第 3 种，两个折点连通（双直角连通）
@param p1 第 1 个点
@param p2 第 2 个点
'''
def TwoCornerLink(p1, p2):
    checkP=Point(p1.x, p1.y)
    #4 个方向的探测开始
    for i in range(0,4):
        checkP.x=p1.x
        checkP.y=p1.y
        #向下
        if(i==3):
            checkP.y+=1
            while((checkP.y<Height) and map[checkP.x][checkP.y]==" "):
                linePointStack.append(checkP)
                if(OneCornerLink(checkP, p2)):
                    print("下探测 OK")
                    return True
                else:
                    linePointStack.pop()
                checkP.y+=1
        #向右
        elif(i==2):
            checkP.x+=1
            while((checkP.x<Width) and map[checkP.x][checkP.y]==" "):
                linePointStack.append(checkP)
                if (OneCornerLink(checkP, p2)):
                    print("右探测 OK")
                    return True
                else:
```

```
                linePointStack.pop()
            checkP.x+=1
    #向左
    elif(i==1):
        checkP.x-=1
        while((checkP.x>=0) and map[checkP.x][checkP.y]==" "):
            linePointStack.append(checkP)
            if(OneCornerLink(checkP, p2)):
                print("左探测 OK")
                return True
            else:
                linePointStack.pop()
            checkP.x-=1
    #向上
    elif(i==0):
        checkP.y-=1
        while((checkP.y >=0) and map[checkP.x][checkP.y]==" "):
            linePointStack.append(checkP)
            if(OneCornerLink(checkP, p2)):
                print("上探测 OK")
                return True
            else:
                linePointStack.pop()
            checkP.y-=1
#4 个方向寻完也没找到适合的 checkP 点
print("两直角连接没找到适合的 checkP 点")
return False;
```

注意：上面的代码在测试两个折点连通时并没有考虑两个折点都在游戏区域外部的情况，有些连连看游戏不允许折点在游戏区域外侧（即边界外）。如果允许这种情况，对上面的代码做如下修改。

```
#向下
if (i==3):
    checkP.y+=1
    while((checkP.y<Height) and map[checkP.x][checkP.y]==" "):
        linePointStack.append(checkP)
        if(OneCornerLink(checkP, p2)):
            print("下探测 OK")
            return True
        else:
            linePointStack.pop()
        checkP.y+=1
    #补充两个折点都在游戏区域底侧外部
    if checkP.y==Height:           #出了底部，仅需判断 p2 能否达到底部边界
        z=Point(p2.x,Height-1)     #底部边界点
```

```
        if lineCheck(z,p2) : #两个折点在区域外部的底侧
            linePointStack.append(Point(p1.x, Height))
            linePointStack.append(Point(p2.x, Height))
            print("下探测到游戏区域外部 OK")
            return True
```

对于其余 3 个方向的边界外部两个折点的连通情况判断，请读者自己思考添加。

❸ 智能查找功能的实现

若要在地图上自动查找出一组相同、可以抵消的方块，通常采用遍历算法。下面借助图 10-6 分析此算法。

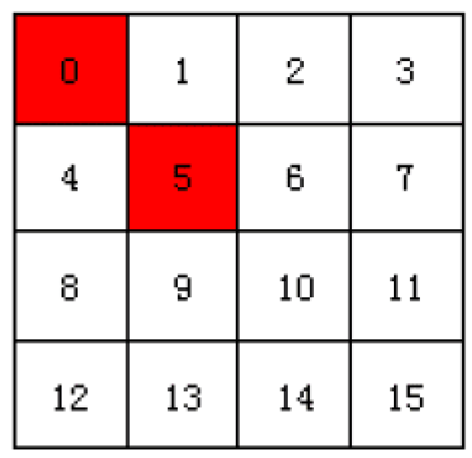
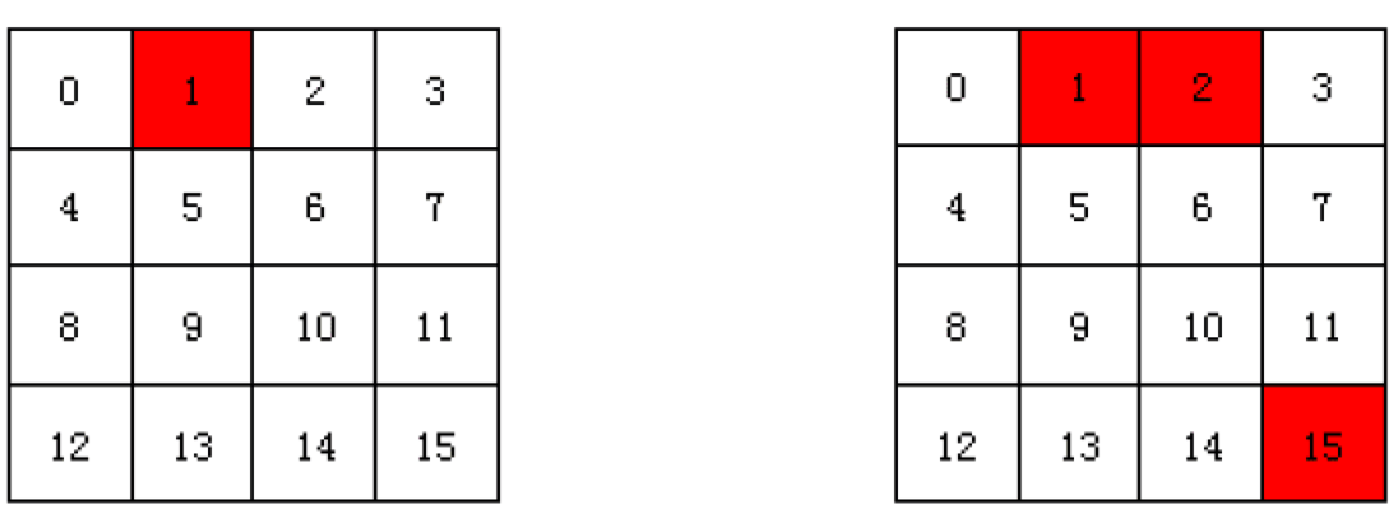


图 10-6　匹配示意图 1

在图中找相同的方块时，将按方块地图 map 的下标位置对每个方块进行查找，一旦找到一组相同、可以抵消的方块则马上返回。在查找相同方块组的时候，必须先确定第 1 个选定方块（例如 0 号方块），然后在这个基础上做遍历查找第 2 个选定方块，即从 1 开始按照 1、2、3、4、5、6、7 等的顺序查找第 2 个选定方块，并判断选定的两个方块是否连通抵消，假如 0 号方块与 5 号方块连通，则经过(0,1)、(0,2)、(0,3)、(0,4)、(0,5)共 5 组数据的判断对比，若成功立即返回。

如果找不到匹配的第 2 个选定方块，则如图 10-7（a）所示编号加 1 重新选定第 1 个方块（即 1 号方块）进入下一轮，然后在这个基础上做遍历查找第 2 个选定方块，即如图 10-7（b）所示从 2 号开始按照 2、3、4、5、6、7 等的顺序查找第 2 个选定方块，直到搜索到最后一块（即 15 号方块）。那么为什么从 2 开始查找第 2 个选定方块，而不是从 0 号开始呢？因为在将 1 号方块选定为第 1 个选定方块前，0 号已经作为第 1 个选定方块对后面的方块进行可连通的判断了，它必然不会与后面的方块连通。



（a）0 号方块找不到匹配方块，选定 1 号　　（b）从 2 号方块开始找匹配

图 10-7　匹配示意图 2

如果找不到与 1 号方块连通且相同的方块，则编号加 1 重新选定第 1 个方块（即 2 号方块）进入下一轮，从 3 号开始按照 3、4、5、6、7 等的顺序查找第 2 个选定方块。

按照上面设计的算法，整个流程图如图 10-8 所示。

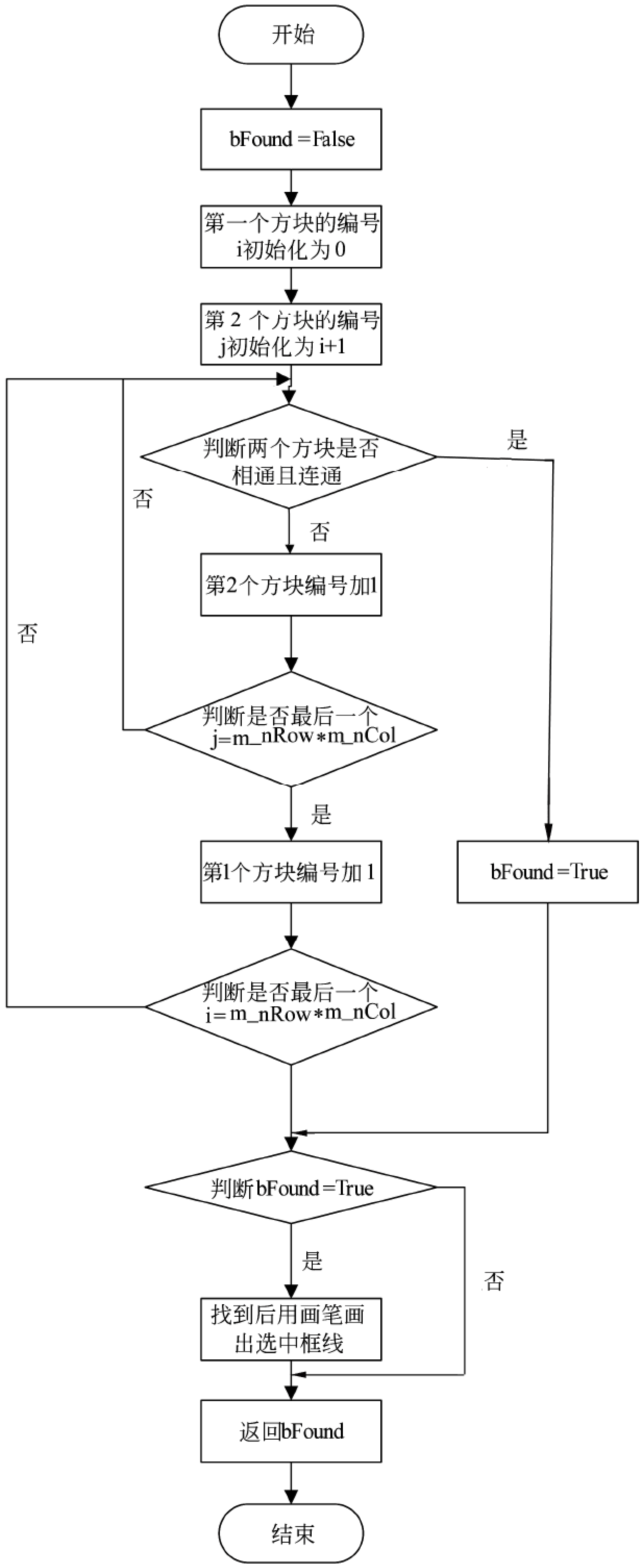


图 10-8　智能查找匹配方块的流程图

根据流程图，把自动查找出一组相同、可以抵消的方块的功能封装在 Find2Block()方法里面，其代码如下：

```
def find2Block(event):                          #自动查找
    global firstSelectRectId,SecondSelectRectId
    m_nRoW=Height
    m_nCol=Width
    bFound=False;
    #第 1 个方块从地图的 0 位置开始
    for i in range(0, m_nRoW* m_nCol):
        #找到则跳出循环
        if(bFound):
            break
        #算出对应的虚拟行/列位置
        x1=i%m_nCol
        y1=i//m_nCol
        p1=Point(x1,y1)
        #无图案的方块跳过
        if (map[x1][y1]==' '):
            continue
        #第 2 个方块从前一个方块的后面开始
        for j in range(i+1, m_nRoW* m_nCol):
            #算出对应的虚拟行/列位置
            x2=j%m_nCol
            y2=j//m_nCol
            p2=Point(x2,y2)
            #第 2 个方块不为空，且与第 1 个方块的图标相同
            if(map[x2][y2]!=' ' and IsSame(p1,p2)):
                #判断是否可以连通
                if (IsLink(p1, p2)):
                    bFound=True;
                    break
    #找到后
    if(bFound):                                 #p1(x1,y1)与p2(x2,y2)连通
        print('找到后',p1.x,p1.y,p2.x,p2.y)
        #画选定(x1,y1)处的框线
        firstSelectRectId=cv.create_rectangle(x1*40,y1*40,x1*40+40,
        y1*40+40,width=2,outline="red")
        #画选定(x2,y2)处的框线
        secondSelectRectId=cv.create_rectangle(x2*40,y2*40,x2*40+40,
        y2*40+40,outline="red")
        #t=Timer(timer_interval,delayrun)    #定时函数自动消除
        #t.start()
    return bFound
```

10.3 关键技术

10.3.1 图形绘制——Tinker 的 Canvas 组件

视频讲解

Tinker 的 Canvas（画布）是一个长方形区域，用于图形绘制或复杂的图形界面布局。用户可以在画布上绘制图形、创建文字、放置各种组件和框架。

❶ 创建 Canvas 对象

用户可以使用下面的方法创建一个 Canvas 对象。

```
Canvas 对象=Canvas(窗口对象, 选项, ... )
```

其常用选项如表 10-1 所示。

表 10-1 Canvas 的常用选项

属　性	说　明
bd	指定画布的边框宽度，单位是像素
bg	指定画布的背景颜色
confine	指定画布在滚动区域外是否可以滚动，默认为 True，表示不能滚动
cursor	指定画布中的鼠标指针，例如 arrow、circle、dot
height	指定画布的高度
highlightcolor	选中画布时的背景色
relief	指定画布的边框样式，可选值有 SUNKEN、RAISED、GROOVE、RIDGE
scrollregion	指定画布的滚动区域的元组(w,n,e,s)

❷ 显示 Canvas 对象

显示 Canvas 对象的方法如下：

```
Canvas 对象.pack()
```

例如创建一个白色背景、宽 300、高 120 的画布。

```
from tkinter import *
root=Tk()
cv=Canvas(root, bg='white', width=300, height=120)
cv.create line(10,10,100,80,width=2, dash=7)    #绘制直线
cv.pack()                                       #显示画布
root.mainloop()
```

10.3.2 Canvas 上的图形对象

在 Canvas 上可以绘制各种图形对象，通过调用以下绘制函数实现。

- create_arc()：绘制圆弧。
- create_line()：绘制直线。
- create_bitmap()：绘制位图。

- create_image()：绘制位图图像。
- create_oval()：绘制椭圆。
- create_rectangle()：绘制矩形。
- create_polygon()：绘制多边形。
- create_window()：绘制子窗口。
- create_text()：创建一个文字对象。

Canvas 上的每个绘制对象都有一个标识 ID（整数），在使用绘制函数创建绘制对象时返回绘制对象 ID。例如：

```
id1=cv.create_line(10,10,100,80,width=2, dash=7)   #绘制直线
```

通过 ID1 可以得到绘制对象直线 ID。

在创建图形对象时可以使用 tags 属性设置图形对象的标记（tag），例如：

```
rt=cv.create_rectangle(10,10,110,110, tags='r1')
```

上面的语句指定矩形对象 rt 具有一个标记 r1。

用户也可以同时设置多个标记，例如：

```
rt=cv.create_rectangle(10,10,110,110, tags=('r1','r2','r3'))
```

上面的语句指定矩形对象 rt 具有 3 个标记 r1、r2、r3。

在指定标记后，使用 find_withtag()方法可以获取到指定 tag 的图形对象，然后设置图形对象的属性。find_withtag()方法的语法如下：

```
Canvas 对象.find_withtag(tag 名)
```

find_withtag()方法返回一个图形对象数组，其中包含所有具有 tag 名的图形对象。

使用 itemconfig()方法可以设置图形对象的属性，语法如下：

```
Canvas 对象.itemconfig(图形对象, 属性 1=值 1, 属性 2=值 2…)
```

例 10-1　使用 tags 属性设置图形对象标记。

```
from tkinter import *
root=Tk()
#创建一个 Canvas，设置其背景色为白色
cv=Canvas(root, bg='white', width=200, height=200)
#使用 tags 给第 1 个矩形指定 3 个 tag
rt=cv.create rectangle(10,10,110,110, tags=('r1','r2','r3'))
cv.pack()
cv.create rectangle(20,20,80,80, tags ='r3')#使用 tags 给第 2 个矩形指定一个 tag
#将所有与 tag('r3')绑定的 item 边框的颜色设置为蓝色
for item in cv.find withtag('r3'):
    cv.itemconfig(item,outline='blue')
root.mainloop()
```

下面学习使用绘制函数绘制各种图形对象及对图形对象进行修改等操作。

❶ 绘制圆弧

使用 create_arc()方法可以创建一个圆弧对象，可以是一个弦、饼图扇区或者是一个简单的弧，具体语法如下：

```
Canvas 对象.create_arc(弧外框矩形左上角的 x 坐标，弧外框矩形左上角的 y 坐标，弧外框矩形右下角的 x 坐标，弧外框矩形右下角的 y 坐标，选项，...)
```

创建圆弧时的常用选项：outline 指定圆弧边框的颜色，fill 指定填充颜色，width 指定圆弧边框的宽度，start 代表起始角度，extent 代表指定角度偏移量而不是终止角度。

例 10-2　使用 create_ arc()方法创建圆弧，运行效果如图 10-9 所示。

```
from tkinter import *
root=Tk()
#创建一个 Canvas，设置其背景色为白色
cv=Canvas(root,bg='white')
cv.create_arc((10,10,110,110),)      #使用默认参数创建一个圆弧，结果为 90° 的扇形
d={1:PIESLICE,2:CHORD,3:ARC}
for i in d:
    #使用 3 种样式，分别创建了扇形、弓形和弧形
    cv.create_arc((10,10+60*i,110,110+60*i),style=d[i])
    print(i,d[i])
#使用 start/extent 指定圆弧起始角度与偏移角度
cv.create arc(
        (150,150,250,250),
        start=10,          #指定起始角度
        extent=120         #指定角度偏移量（逆时针）
        )
cv.pack()
root.mainloop()
```

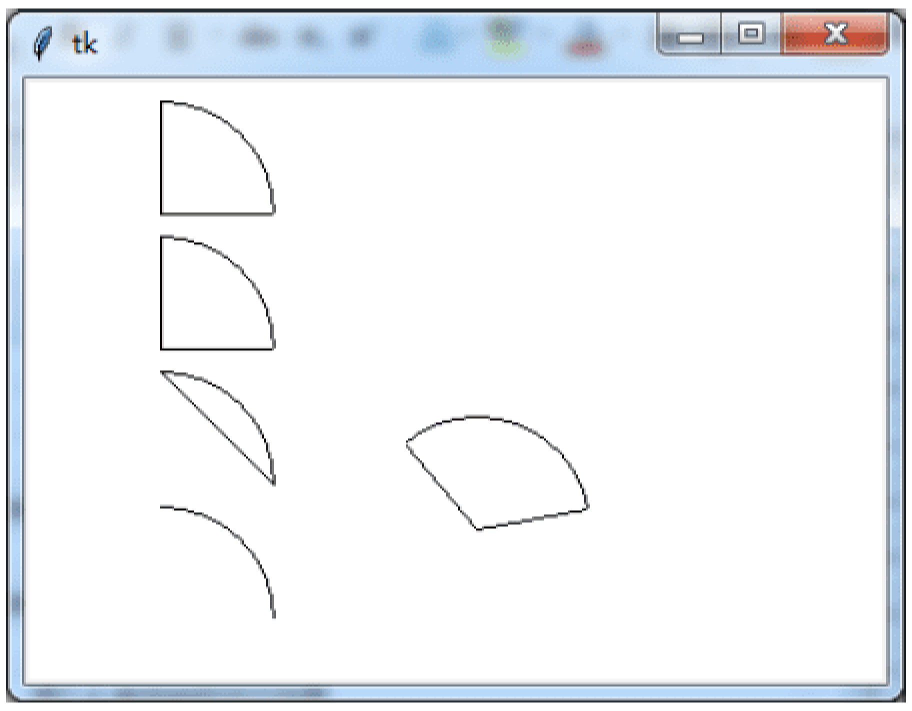


图 10-9　创建圆弧

❷ 绘制线条

使用 create_line()方法可以创建一个线条对象，具体语法如下：

```
line=canvas.create_line(x0, y0, x1, y1, …, xn, yn, 选项)
```

参数（x0, y0），（x1, y1），…，（xn, yn）是线段的端点。

创建线段时的常用选项：width 指定线段宽度，arrow 指定是否使用箭头（没有箭头为 none，起点有箭头为 first，终点有箭头为 last，两端有箭头为 both），fill 指定线段颜色，dash 指定线段为虚线（其整数值决定虚线的样式）。

例 **10-3**　使用 create_line()方法创建线条，运行效果如图 10-10 所示。

```
from tkinter import *
root=Tk()
cv=Canvas(root, bg='white', width=200, height=100)
cv.create_line(10, 10, 100, 10, arrow='none')          #绘制没有箭头的线段
cv.create line(10, 20, 100, 20, arrow='first')         #绘制起点有箭头的线段
cv.create line(10, 30, 100, 30, arrow='last')          #绘制终点有箭头的线段
cv.create line(10, 40, 100, 40, arrow='both')          #绘制两端有箭头的线段
cv.create line(10,50,100,100,width=3, dash=7)          #绘制虚线
cv.pack()
root.mainloop()
```

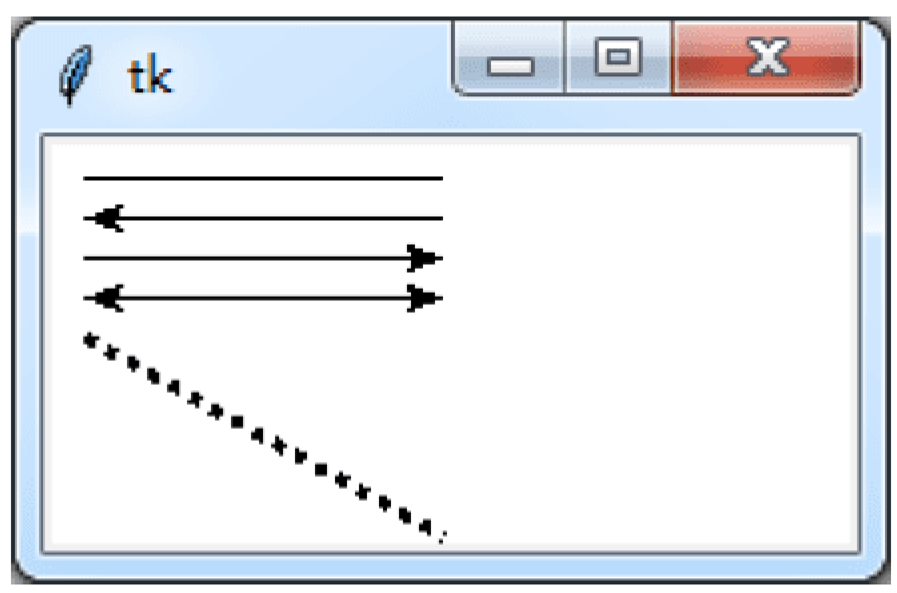


图 10-10　创建线条

❸ 绘制矩形

使用 create_ rectangle()方法可以创建矩形对象，具体语法如下：

```
Canvas 对象.create_rectangle(矩形左上角的 x 坐标, 矩形左上角的 y 坐标, 矩形右下角
的 x 坐标, 矩形右下角的 y 坐标, 选项, ...)
```

创建矩形对象时的常用选项：outline 指定边框颜色，fill 指定填充颜色，width 指定边框的宽度，dash 指定边框为虚线，stipple 使用指定自定义画刷填充矩形。

例 **10-4**　使用 create_rectangle()方法创建矩形，运行效果如图 10-11 所示。

```
from tkinter import *
root=Tk()
#创建一个 Canvas，设置其背景色为白色
cv=Canvas(root, bg='white', width=200, height=100)
cv.create rectangle(10,10,110,110, width =2,fill='red')
                                             #指定矩形的填充色为红色、宽度为 2
cv.create_rectangle(120, 20, 180, 80, outline='green')#指定矩形的边框颜色为绿色
cv.pack()
root.mainloop()
```

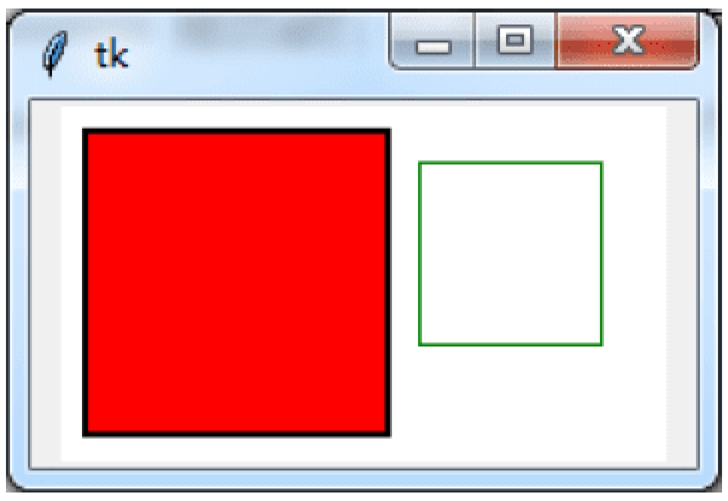


图 10-11　创建矩形对象

❹ 绘制多边形

使用 create_polygon()方法创建一个多边形对象，可以是一个三角形、矩形或者任意一个多边形，具体语法如下：

```
Canvas 对象. create polygon(顶点 1 的 x 坐标, 顶点 1 的 y 坐标, 顶点 2 的 x 坐标, 顶点 2 的 y 坐标, …, 顶点 n 的 x 坐标, 顶点 n 的 y 坐标, 选项, …)
```

创建多边形对象时的常用选项：outline 指定边框颜色，fill 指定填充颜色，width 指定边框的宽度，smooth 指定多边形的平滑程度（等于 0 表示多边形的边是折线，等于 1 表示多边形的边是平滑曲线）。

例 10-5　创建三角形、正方形、对顶三角形，运行效果如图 10-12 所示。

```
from tkinter import *
root=Tk()
cv=Canvas(root, bg='white', width=300, height=100)
cv.create polygon(35,10,10,60,60,60, outline='blue',fill='red', width=2)
                                                            #等腰三角形
cv.create_polygon(70,10,120,10,120,60,outline='blue',fill='white',width=2)
                                                            #直角三角形
cv.create polygon (130,10,180,10,180,60,130,60, width=4)   #黑色填充正方形
cv.create polygon (190,10,240,10,190,60,240,60, width=1)   #对顶三角形
cv.pack()
root.mainloop()
```

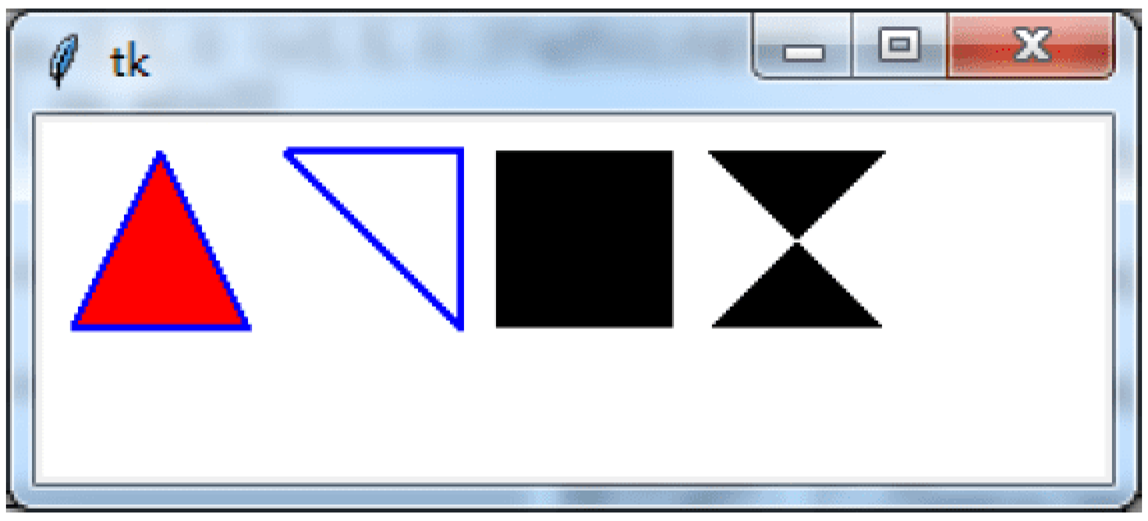


图 10-12　创建三角形、正方形、对顶三角形

❺ 绘制椭圆

使用 create_oval()方法可以创建一个椭圆对象，具体语法如下：

```
Canvas 对象.create_oval(包裹椭圆的矩形左上角 x 坐标, 包裹椭圆的矩形左上角 y 坐标, 包
```

```
裹椭圆的矩形右下角 x 坐标, 包裹椭圆的矩形右下角 y 坐标, 选项, ...)
```

创建椭圆对象时的常用选项：outline 指定边框颜色，fill 指定填充颜色，width 指定边框的宽度。如果包裹椭圆的矩形是正方形，则绘制的是一个圆形。

例 10-6　创建椭圆和圆形，运行效果如图 10-13 所示。

```
from tkinter import *
root=Tk()
cv=Canvas(root, bg='white', width=200, height=100)
cv.create oval(10,10,100,50, outline='blue', fill='red', width=2)#椭圆
cv.create oval(100,10,190,100, outline='blue', fill='red', width=2)#圆形
cv.pack()
root.mainloop()
```

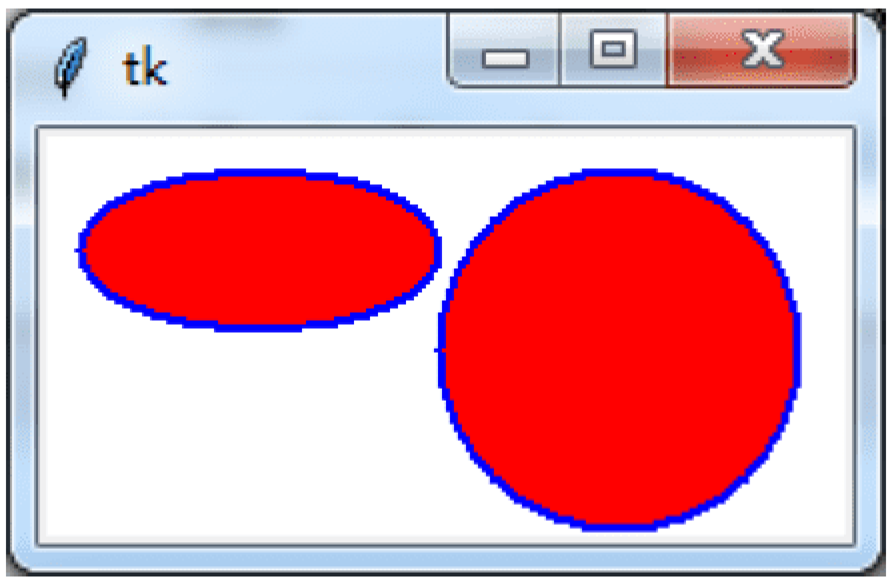


图 10-13　创建椭圆和圆形

❻ 创建文字

使用 create_text()方法可以创建一个文字对象，具体语法如下：

```
文字对象=Canvas 对象.create_text((文本左上角的 x 坐标,文本左上角的 y 坐标), 选项, …)
```

创建文字对象时的常用选项：text 是文字对象的文本内容，fill 指定文字颜色，anchor 控制文字对象的位置（其取值'w'表示左对齐，'e'表示右对齐，'n'表示顶对齐，'s'表示底对齐，'nw'表示左上对齐，'sw'表示左下对齐，'se'表示右下对齐，'ne'表示右上对齐，'center'表示居中对齐，anchor 的默认值为'center'），justify 设置文字对象中文本的对齐方式（其取值'left'表示左对齐，'right'表示右对齐，'center'表示居中对齐，justify 的默认值为'center'）。

例 10-7　创建文本，运行效果如图 10-14 所示。

```
from tkinter import *
root=Tk()
cv=Canvas(root, bg='white', width=200, height=100)
cv.create text((10,10), text='Hello Python', fill='red', anchor='nw')
cv.create_text((200,50), text='你好, Python', fill='blue', anchor='se')
cv.pack()
root.mainloop()
```

select_from()方法用于指定选中文本的起始位置，具体语法如下：

```
Canvas 对象.select_from(文字对象, 选中文本的起始位置)
```

select_to()方法用于指定选中文本的结束位置，具体语法如下：

```
Canvas 对象.select_to(文字对象, 选中文本的结束位置)
```

例 10-8　选中文本的例子，运行效果如图 10-15 所示。

```
from tkinter import *
root=Tk()
cv=Canvas(root, bg='white', width=200, height=100)
txt=cv.create_text((10,10), text='中原工学院计算机学院', fill='red', anchor='nw')
#设置选中文本的起始位置
cv.select_from(txt,5)
#设置选中文本的结束位置
cv.select_to(txt,9)                    #选中“计算机学院”
cv.pack()
root.mainloop()
```

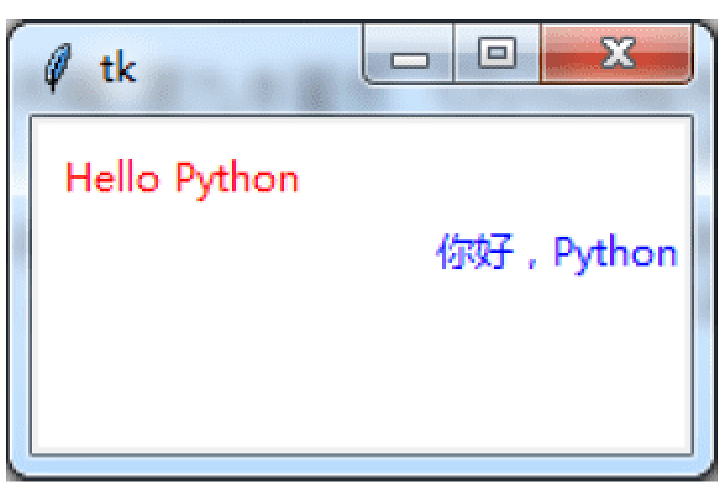


图 10-14　创建文本

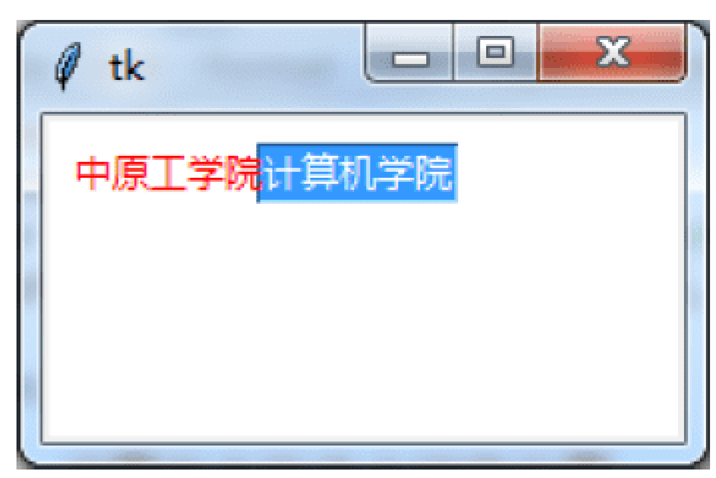


图 10-15　选中文本

❼ 绘制位图和图像

1）绘制位图

使用 create_bitmap()方法可以绘制 Python 内置的位图，具体语法如下：

```
Canvas 对象.create_bitmap((x 坐标,y 坐标),bitmap=位图字符串, 选项, …)
```

(x 坐标,y 坐标)是位图放置的中心坐标。常用选项：bitmap、activebitmap 和 disabledbitmap 分别用于指定正常、活动、禁用状态显示的位图。

2）绘制图像

在游戏开发中需要使用大量图像，采用 create_bitmap()方法可以绘制图形图像，具体语法如下：

```
Canvas 对象.create_image((x 坐标,y 坐标), image=图像文件对象, 选项, …)
```

(x 坐标,y 坐标)是图像放置的中心坐标。常用选项：image、activeimage 和 disabled image 用于指定正常、活动、禁用状态显示的图像。

注意：使用 PhotoImage()函数来获取图像文件对象。

```
img1=PhotoImage(file=图像文件)
```

例如，img1=PhotoImage(file='C:\\aa.png') 获取笑脸图形。Python 支持的图像文件格式一般为.png 和.gif。

例 10-9　绘制图像示例，运行效果如图 10-16 所示。

```
from tkinter import *
root=Tk()
cv=Canvas(root)
img1=PhotoImage(file='C:\\aa.png')                  #笑脸
img2=PhotoImage(file='C:\\2.gif')                   #方块 A
img3=PhotoImage(file='C:\\3.gif')                   #梅花 A
cv.create image((100,100),image=img1)               #绘制笑脸
cv.create_image((200,100),image=img2)               #绘制方块 A
cv.create_image((300,100),image=img3)               #绘制梅花 A
d={1:'error',2:'info',3:'question',4:'hourglass',5:'questhead',
    6:'warning',7:'gray12',8:'gray25',9:'gray50',10:'gray75'} #字典
#cv.create bitmap((10,220),bitmap=d[1])
#以下遍历字典绘制 Python 内置的位图
for i in d:
    cv.create_bitmap((20*i,20),bitmap=d[i])
cv.pack()
root.mainloop()
```

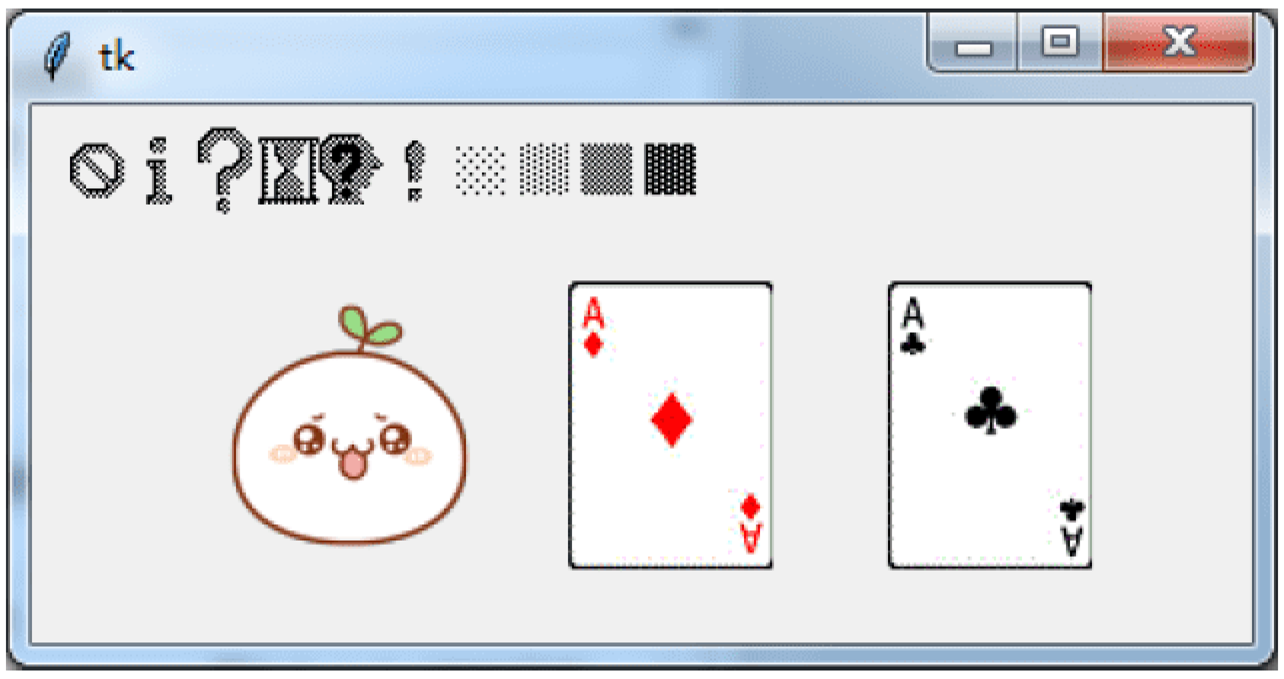


图 10-16　绘制图像示例

在学会绘制图像之后，就可以开发图形版的扑克牌、连连看、推箱子等游戏了。

❽ 修改图形对象的坐标

使用 coords()方法可以修改图形对象的坐标，具体语法如下：

```
Canvas 对象.coords(图形对象，(图形左上角的 x 坐标，图形左上角的 y 坐标，图形右下角的 x
坐标，图形右下角的 y 坐标))
```

因为可以同时修改图形对象的左上角的坐标和右下角的坐标，所以可以缩放图形对象。

注意： 如果图形对象是图像文件，则只能指定图像的中心点坐标，而不能指定图像对象左上角的坐标和右下角的坐标，故不能缩放图像。

例 10-10 修改图形对象的坐标示例，运行效果如图 10-17 所示。

```
from tkinter import *
root=Tk()
cv=Canvas(root)
img1=PhotoImage(file='C:\\aa.png')                    #笑脸
img2=PhotoImage(file='C:\\2.gif')                     #方块 A
img3=PhotoImage(file='C:\\3.gif')                     #梅花 A
rt1=cv.create_image((100,100),image=img1)             #绘制笑脸
rt2=cv.create_image((200,100),image=img2)             #绘制方块 A
rt3=cv.create_image((300,100),image=img3)             #绘制梅花 A
#重新设置方块 A（rt2 对象）的坐标
cv.coords(rt2,(200,50))                               #调整 rt2 对象的位置
rt4=cv.create_rectangle(20,140,110,220,outline='red',fill='green')#正方形对象
cv.coords(rt4,(100,150,300,200))                      #调整 rt4 对象的位置
cv.pack()
root.mainloop()
```

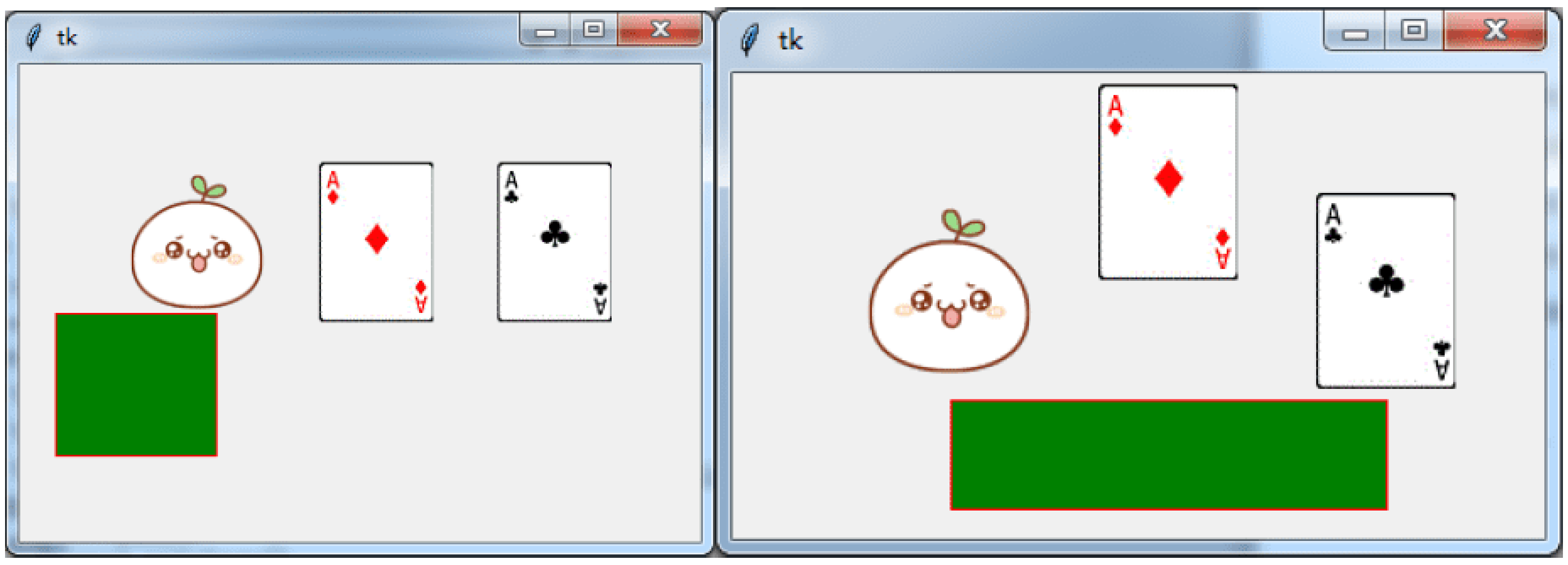


图 10-17　调整图形对象位置之前和之后的效果

❾ 移动指定图形对象

使用 move()方法可以移动图形对象的坐标，具体语法如下：

```
Canvas 对象.move(图形对象, x 坐标偏移量, y 坐标偏移量)
```

例 10-11 移动指定图形对象示例，运行效果如图 10-18 所示。

```
from tkinter import *
root=Tk()
#创建一个 Canvas，设置其背景色为白色
cv=Canvas(root, bg='white', width=200, height=120)
rt1=cv.create rectangle(20,20,110,110,outline='red',stipple='gray12',
fill='green')
cv.pack()
rt2=cv.create_rectangle(20,20,110,110,outline='blue')
```

```
cv.move(rt1,20,-10)      #移动 rt1
cv.pack()
root.mainloop()
```

为了对比移动图形对象的效果，程序在同一位置绘制了两个矩形，其中矩形 rt1 有背景花纹，矩形 rt2 无背景填充，然后用 move()方法移动 rt1，将被填充的矩形 rt1 向右移动 20 像素、向上移动 10 像素，则出现如图 10-18 所示的效果。

⑩ 删除图形对象

使用 delete ()方法可以删除图形对象，具体语法如下：

```
Canvas 对象.delete(图形对象)
```

例如：

```
cv.delete(rt1)            #删除 rt1 图形对象
```

⑪ 缩放图形对象

使用 scale()方法可以缩放图形对象，具体语法如下：

```
Canvas 对象.scale(图形对象,X 轴偏移量,Y 轴偏移量,X 轴缩放比例,Y 轴缩放比例)
```

例 10-12　缩放图形对象示例，对相同图形对象放大、缩小，运行效果如图 10-19 所示。

```
from tkinter import *
root=Tk()
#创建一个 Canvas，设置其背景色为白色
cv=Canvas(root, bg='white', width=200, height=300)
rt1=cv.create rectangle(10,10,110,110,outline='red',stipple='gray12',
fill='green')
rt2=cv.create rectangle(10,10,110,110,outline='green',stipple='gray12',
fill='red')
cv.scale(rt1,0,0,1,2)               #y 方向放大一倍
cv.scale(rt2,0,0,0.5,0.5)           #缩小一半大小
cv.pack()
root.mainloop()
```

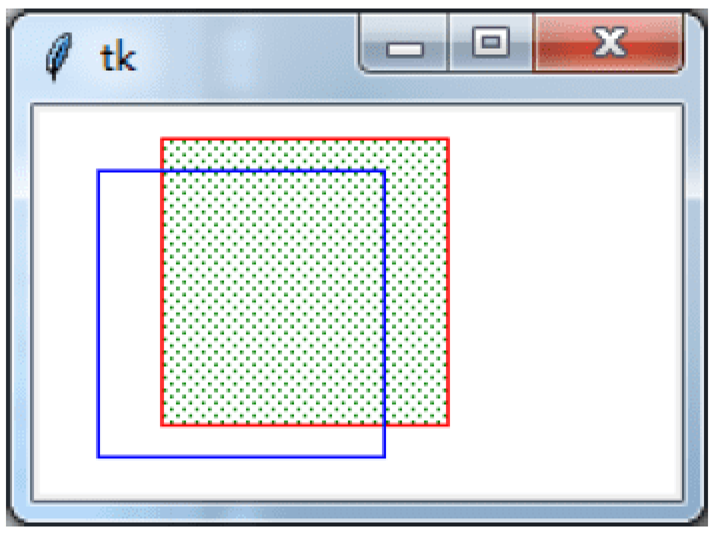


图 10-18　移动指定图形对象运行效果

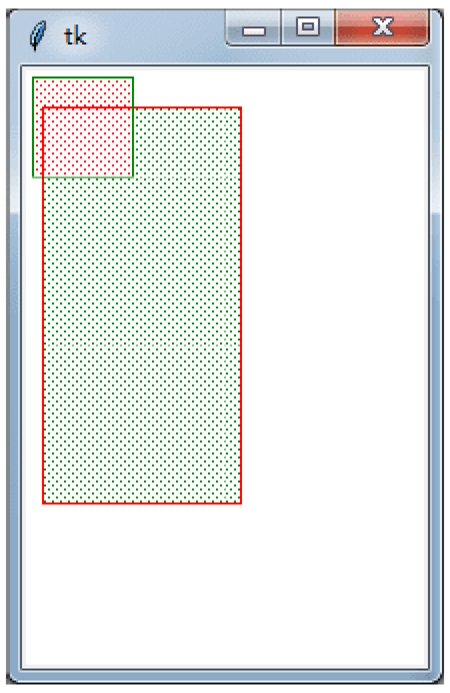


图 10-19　缩放图形对象运行效果

10.4 程序设计的步骤

视频讲解

❶ 设计点类 Point

点类 Point 比较简单，主要存储方块所在棋盘的坐标(x, y)。

```
class Point:    #点类
    def __init__(self,x,y):
        self.x=x
        self.y=y
```

❷ 设计游戏主逻辑

整个游戏在 Canvas 对象中，调用 create_map()实现将图标图案随机放到地图中，地图 map 中记录的是图案的数字编号，最后调用 print_map()按地图 map 中记录的图案信息将图 10-2 中的图标图案绘制在 Canvas 对象中，生成游戏开始的界面，同时绑定 Canvas 对象的鼠标左键和右键事件，并进入窗体显示线程中。

```
root=Tk()
root.title("python 连连看")
imgs=[PhotoImage(file='H:\\ 连 连 看 \\gif\\bar 0'+str(i)+'.gif')  for  i  in
range(0,10)]                          #所有图标图案
Select_first=False                    #是否已经选中第 1 块
firstSelectRectId=-1                  #被选中第 1 块地图对象
SecondSelectRectId=-1                 #被选中第 2 块地图对象
linePointStack=[]                     #存储连接的折点棋盘坐标
Line id=[]
Height=9
Width=10
map=[[" " for y in range(Height)]for x in range(Width)]
image map=[[" " for y in range(Height)]for x in range(Width)]
cv=Canvas(root, bg='green', width=610, height=610)
cv.bind("<Button-1>", callback)       #鼠标左键事件
cv.bind("<Button-3>", find2Block)     #鼠标右键事件
cv.pack()
create_map()                          #产生 map 地图
print map()                           #打印 map 地图
root.mainloop()
```

❸ 编写函数代码

print_map()按地图 map 中记录的图案信息将图 10-2 中的图标图案显示在 Canvas 对象中，生成游戏开始的界面。

```
def print_map():                      #输出 map 地图
    global image map
    for x in range(0,Width):
        for y in range(0,Height):
```

```
            if(map[x][y]!=' '):
                img1=imgs[int(map[x][y])]
                id=cv.create_image((x*40+20,y*40+20),image=img1)
                image_map[x][y]=id
    cv.pack()
    for y in range(0,Height):
        for x in range(0,Width):
            print(map[x][y],end=' ')
        print(",",y)
```

用户在窗口中单击时，由屏幕像素坐标(event.x,event.y)计算被单击方块的地图棋盘位置坐标(x,y)。判断是否为第 1 次选中方块，是则仅仅对选定方块加上蓝色的示意框线。如果是第 2 次选中方块，则加上黄色的示意框线，同时要判断是否图案相同且连通。假如连通则画选中方块之间的连接线，延时 0.3 秒后清除第 1 个选定方块和第 2 个选定方块图案，并清除选中方块之间的连接线。假如不连通，则是清除选定两个方块示意框线。

Canvas 对象鼠标右键事件调用智能查找功能 Find2Block()。

```
def find2Block(event):  #自动查找
        …               #见前面程序设计的思路
```

Canvas 对象鼠标左键事件代码：

```
def callback(event):   #鼠标左键事件代码
    global Select_first,p1,p2
    global firstSelectRectId,SecondSelectRectId
    #print("clicked at", event.x, event.y,turn)
    x=(event.x)//40      #换算棋盘坐标
    y=(event.y)//40
    print("clicked at", x, y)

    if map[x][y]==" ":
       showinfo(title="提示",message="此处无方块")
    else:
       if Select_first==False:
           p1=Point(x,y)
           #画选定(x1,y1)处的框线
           firstSelectRectId=cv.create_rectangle(x*40,y*40,x*40+40,y*40+
           40,outline="blue")
           Select_first=True
       else:
           p2=Point(x,y)
           #判断第 2 次单击的方块是否已被第 1 次单击选取，如果是则返回
           if (p1.x==p2.x) and (p1.y==p2.y):
               return
           #画选定(x2,y2)处的框线
           print('第 2 次单击的方块',x,y)
```

```
            SecondSelectRectId=cv.create_rectangle(x*40,y*40,x*40+40,y*40
            +40,outline="yellow")
            print('第 2 次单击的方块',SecondSelectRectId)
            cv.pack()
            if IsSame(p1,p2) and IsLink(p1,p2):  #判断是否连通
                print('连通',x,y)
                Select first=False
                #画选中方块之间的连接线
                drawLinkLine(p1,p2)
                t=Timer(timer interval,delayrun)  #定时函数
                t.start()
            else:                                 #不能连通则取消选定的两个方块
                cv.delete(firstSelectRectId)      #清除第 1 个选定框线
                cv.delete(SecondSelectRectId)     #清除第 2 个选定框线
                Select_first=False
```

IsSame(p1,p2)判断 p1 (x1, y1)和 p2(x2, y2)处的方块图案是否相同。

```
def IsSame(p1,p2):
    if map[p1.x][p1.y]==map[p2.x][p2.y]:
        print("clicked at IsSame")
        return True
    return False
```

以下是画方块之间的连接线和清除连接线的方法。

drawLinkLine(p1,p2)绘制(p1,p2)所在两个方块之间的连接线。判断 linePointStack 列表长度，如果为 0，则是直接连通；linePointStack 列表长度为 1，则是一折连通，linePointStack 存储的是一折连通的折点；linePointStack 列表长度为 2，则是两折连通，linePointStack 存储的是两折连通的两个折点。

```
def drawLinkLine(p1,p2):       #画连接线
    if(len(linePointStack)==0):
        Line id.append(drawLine(p1,p2))
    else:
        print(linePointStack,len(linePointStack))
    if(len(linePointStack)==1):
        z=linePointStack.pop()
        print("一折连通点 z",z.x,z.y)
        Line id.append(drawLine(p1,z))
        Line id.append(drawLine(p2,z))
    if(len(linePointStack)==2):
        z1=linePointStack.pop()
        print("两折连通点 z1",z1.x,z1.y)
        Line id.append(drawLine(p2,z1))
        z2=linePointStack.pop()
        print("两折连通点 z2",z2.x,z2.y)
```

```
            Line_id.append(drawLine(z1,z2))
            Line_id.append(drawLine(p1,z2))
```

drawLinkLine(p1,p2)绘制(p1,p2)之间的直线。

```
def drawLine(p1,p2):
    print("drawLine p1,p2",p1.x,p1.y,p2.x,p2.y)
    id=cv.create_line(p1.x*40+20,p1.y*40+20,p2.x*40+20,p2.y*40+20,width=5,
    fill='red')
    #cv.pack()
    return id
```

undrawConnectLine()删除 Line_id 记录的连接线。

```
def undrawConnectLine():
    while len(Line_id)>0:
        idpop=Line_id.pop()
        cv.delete(idpop)
```

clearTwoBlock()清除(p1, p2)之间的连线及所在方块的图案。

```
def clearTwoBlock():          #清除连线及方块
    #清除第 1 个选定框线
    cv.delete(firstSelectRectId)
    #清除第 2 个选定框线
    cv.delete(SecondSelectRectId)
    #清空记录方块的值
    map[p1.x][p1.y]=" "
    cv.delete(image_map[p1.x][p1.y])
    map[p2.x][p2.y]=" "
    cv.delete(image_map[p2.x][p2.y])
    Select_first=False
    undrawConnectLine()       #清除选中方块之间的连接线
```

delayrun()函数是定时函数，延时 timer_interval（0.3 秒）后清除(p1,p2)之间的连线及所在方块的图案。

```
timer_interval=0.3            #0.3 秒
def delayrun():
    clearTwoBlock()           #清除连线及方块
```

IsWin()检测是否尚有未被消除的方块，即地图 map 中的元素值非空（" "），如果没有则表示已经赢得了游戏。

```
'''
#检测是否已经赢得了游戏
'''
def IsWin()
    #检测是否尚有未被消除的方块
```

```
#(非 BLANK_STATE 状态)
for y in range(0,Height):
  for x in range(0,Width):
     if map[i]!=" "):
       return False;
return True;
```

至此完成连连看游戏。

第11章

益智游戏——推箱子游戏

视频讲解

11.1　推箱子游戏介绍

经典的推箱子游戏是一个来自日本的古老游戏，目的是训练玩家的逻辑思考能力。该游戏的思想是，在一个狭小的仓库中，要求把木箱放到指定的位置，玩家稍不小心就会出现箱子无法移动或者通道被堵住的情况，所以需要巧妙地利用有限的空间和通道合理安排移动的次序和位置，这样才能顺利地完成任务。

推箱子游戏的功能如下：

游戏运行载入相应的地图，屏幕中出现一个推箱子的工人，其周围是围墙、人可以走的通道、几个可以移动的箱子和箱子放置的目的地。让玩家通过按上、下、左、右键控制工人推箱子，当箱子都推到了目的地后出现过关信息，并显示下一关。如果推错了，玩家按空格键重新玩过这关，直到过完全部关卡。

本章开发推箱子游戏，推箱子游戏界面如图 11-1 所示。

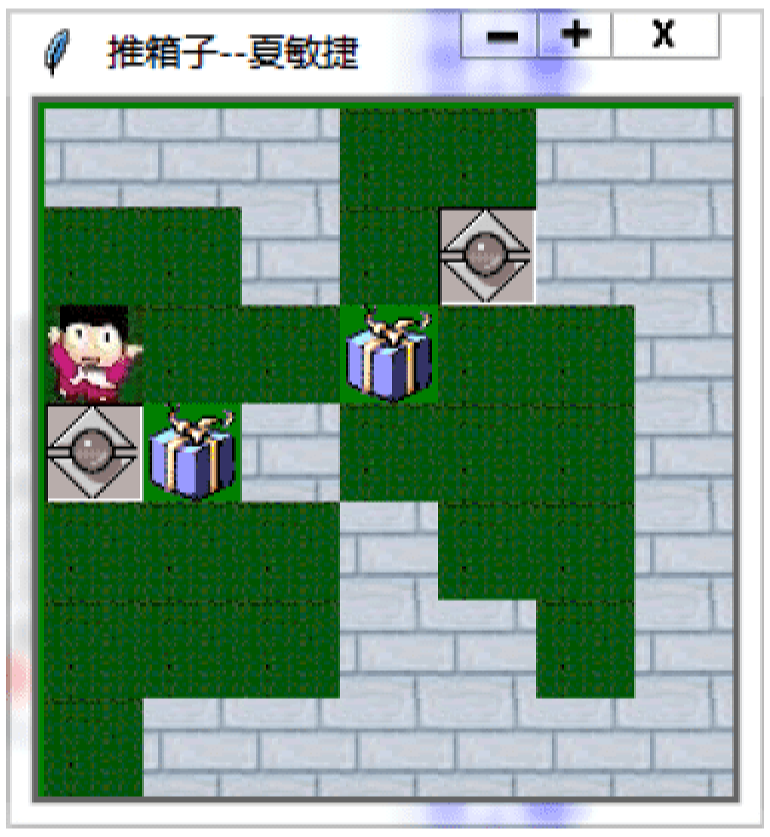
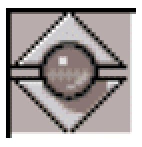
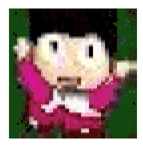


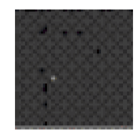
图 11-1　推箱子游戏界面

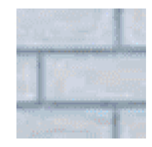
本游戏使用的图片元素的含义如下：

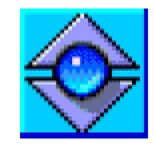

目的地


工人


箱子


通道


围墙


箱子已在目的地

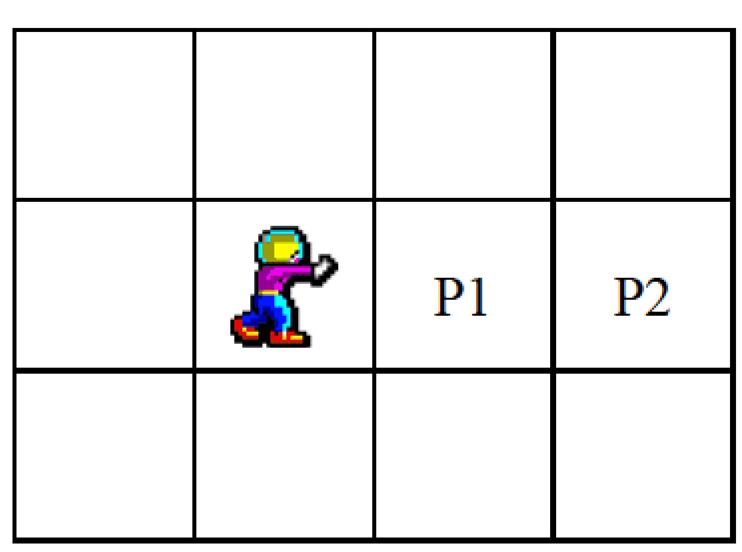
11.2 程序设计的思路

首先来确定一下开发难点。对工人的操作很简单，就是 4 个方向移动，工人移动，箱子也移动，所以对按键的处理也比较简单。当箱子到达目的地位置时就会产生游戏过关事件，需要一个逻辑判断。仔细地想一下，所有事件都发生在一张地图中。这张地图包括了箱子的初始化位置、箱子最终放置的位置和围墙障碍等。每一关地图都要更换，这些位置也要变。所以每关的地图数据是最关键的，它决定了每关的不同场景和物体位置。那么下面重点分析一下地图。

这里把地图想象成一个网格，每个格子就是工人每次移动的步长，也是箱子移动的距离，这样问题就简化多了。首先设计一个 7×7 的二维列表 myArray，按照这样的框架来思考。对于格子的 x、y 两个屏幕像素坐标，可以由二维列表下标换算。

对于每个格子的状态值，分别用常量 Wall（0）代表墙、Worker（1）代表人、Box（2）代表箱子、Passageway（3）代表路、Destination（4）代表目的地、WorkerInDest（5）代表人在目的地、RedBox（6）代表放到目的地的箱子。在文件中，原始地图中格子的状态值采用相应的整数形式存放。

在玩家通过键盘控制工人推箱子的过程中，需要按游戏规则判断是否响应该按键指示。下面分析一下工人将会遇到什么情况，以便归纳出所有的规则和对应算法。为了描述方便，可以假设工人的移动趋势方向为向右，其他方向的原理是一致的。P1、P2 分别代表工人移动趋势方向前的两个方格。



❶ 前方 P1 是通道

如果工人前方是通道，工人可以进到 P1 方格，修改相关位置格子的状态值。

❷ 前方 P1 是围墙或出界

如果工人前方是围墙或出界（即阻挡工人的路线），退出规则判断，布局不做任何改变。

❸ 前方 P1 是目的地

如果工人前方是目的地，工人可以进到 P1 方格，修改相关位置格子的状态值。

❹ **前方 P1 是箱子**



在前面 3 种情况中，只要根据前方 P1 处的物体就能判断出工人是否可以移动，而在第 4 种情况中，需要判断箱子前方 P2 处的物体才能判断出工人是否可以移动，此时有以下可能。

（1）P1 处为箱子，P2 处为墙或出界：如果工人前方 P1 处为箱子，P2 处为墙或出界，退出规则判断，布局不做任何改变。

（2）P1 处为箱子，P2 处为通道：如果工人前方 P1 处为箱子，P2 处为通道，工人可以进到 P1 方格；P2 方格状态为箱子，修改相关位置格子的状态值。

（3）P1 处为箱子，P2 处为目的地：如果工人前方 P1 处为箱子，P2 处为目的地，工人可以进到 P1 方格，P2 方格状态为放置好的箱子，修改相关位置格子的状态值。

（4）P1 处为放到目的地的箱子，P2 处为通道：如果工人前方 P1 处为放到目的地的箱子，P2 处为通道，工人可以进到 P1 方格；P2 方格状态为箱子，修改相关位置格子的状态值。

（5）P1 处为放到目的地的箱子，P2 处为目的地：如果工人前方 P1 处为放到目的地的箱子，P2 处为目的地，工人可以进到 P1 方格；P2 方格状态为放置好的箱子，修改相关位置格子的状态值。

综合前面的分析，可以设计出整个游戏的实现流程。

11.3　关键技术

在该游戏中设计“重玩”功能便于玩家无法通过时重玩此关游戏，这时需要将地图信息恢复到初始状态，所以需要将 7×7 的二维列表 myArray 进行复制，注意此时需要了解“列表复制——深复制”问题。

下面举个例子。

问题描述：已知一个列表 a，生成一个新的列表 b，列表元素是对原列表的复制。

```
a=[1,2]
b=a
```

这种做法其实并未真正生成一个新的列表，b 指向的仍然是 a 所指向的对象。这样，如果对 a 或 b 的元素进行修改，a、b 列表的值同时发生变化。

解决的方法如下：

```
a=[1,2]
b=a[:]            #切片，或者使用 b=copy.copy(a)
```

这样修改 a 对 b 没有影响，修改 b 对 a 没有影响。

这种方法只适用于简单列表，也就是列表中的元素都是基本类型，如果列表元素中还存在列表，这种方法就不适用了。原因就是a[:]这种处理只是将列表元素的值生成一个新的列表，如果列表元素也是一个列表，例如 a=[1,[2]]，那么这种复制对于元素[2]的处理只是复制[2]的引用，并未生成[2]的一个新的列表复制。为了证明这一点，测试步骤如下：

```
>>> a=[1,[2]]
>>> b=a[:]
>>> b
[1, [2]]
>>> a[1].append(3)
>>> a
[1, [2, 3]]
>>> b
[1, [2, 3]]
```

可见，对 a 的修改影响到了 b。如果要解决这一问题，可以使用 copy 模块中的 deepcopy() 函数。修改测试如下：

```
>>> import copy
>>> a=[1,[2]]
>>> b=copy.deepcopy(a)
>>> b
[1, [2]]
>>> a[1].append(3)
>>> a
[1, [2, 3]]
>>> b
[1, [2]]
```

知道这一点是非常重要的，因为在本游戏中需要一个新的二维列表（现在状态地图），并且对这个新的二维列表进行操作，同时不想影响原来的二维列表（原始地图）。

11.4 程序设计的步骤

❶ 设计游戏地图

整个游戏在 7×7 区域中，使用二维列表 myArray 存储。其中，方格状态值 0 代表墙，1 代表人，2 代表箱子，3 代表路，4 代表目的地，5 代表人在目的地，6 代表放到目的地的箱子。例如图 11-1 所示的推箱子游戏界面的对应数据如下：

视频讲解

0	0	0	3	3	0	0
3	3	0	3	4	0	0
1	3	3	2	3	3	0
4	2	0	3	3	3	0
3	3	3	0	3	3	0
3	3	3	0	0	3	0
3	0	0	0	0	0	0

方格状态值采用 myArray1 存储（注意按列存储）：

```
#原始地图
myArray1=[[0,3,1,4,3,3,3],
          [0,3,3,2,3,3,0],
          [0,0,3,0,3,3,0],
          [3,3,2,3,0,0,0],
          [3,4,3,3,3,0,0],
          [0,0,3,3,3,3,0],
          [0,0,0,0,0,0,0]]
```

为了明确表示方格状态信息，这里定义变量名（Python 没有枚举类型）来表示，并使用 imgs 列表存储图像，而且按照图形代号的顺序储存图像。

```
Wall=0
Worker=1
Box=2
Passageway=3
Destination=4
WorkerInDest=5
RedBox=6
#原始地图
myArray1=[[0,3,1,4,3,3,3],
          [0,3,3,2,3,3,0],
          [0,0,3,0,3,3,0],
          [3,3,2,3,0,0,0],
          [3,4,3,3,3,0,0],
          [0,0,3,3,3,3,0],
          [0,0,0,0,0,0,0]]
imgs=[PhotoImage(file='bmp\\Wall.gif'),
      PhotoImage(file='bmp\\Worker.gif'),
      PhotoImage(file='bmp\\Box.gif'),
      PhotoImage(file='bmp\\Passageway.gif'),
      PhotoImage(file='bmp\\Destination.gif'),
      PhotoImage(file='bmp\\WorkerInDest.gif'),
      PhotoImage(file='bmp\\RedBox.gif')]
```

❷ 绘制整个游戏区域图形

绘制整个游戏区域图形就是按照地图 myArray 储存图形代号，从 imgs 列表获取对应图像，显示到 Canvas 上。全局变量 x、y 代表工人当前位置，即(x,y)，从地图 myArray 读取时如果是 1（Worker 值为 1），则记录当前位置。

```
def drawGameImage():
    global x,y
    for i in range(0,7) :                       #0～6
       for j in range(0,7) :                    #0～6
            if myArray[i][j]==Worker :
               x=i                              #工人当前位置(x,y)
               y=j
               print("工人当前位置:",x,y)
            img1=imgs[myArray[i][j]]            #从 imgs 列表获取对应图像
            cv.create_image((i*32+20,j*32+20),image=img1) #显示到 Canvas 上
            cv.pack()
```

❸ 按键事件处理

在游戏中对于用户按键的操作采用 Canvas 对象的 KeyPress 按键事件处理。KeyPress 按键处理函数 callback()根据用户的按键消息计算出工人移动趋势方向前两个方格的位置坐标(x1, y1)、(x2, y2)，将所有位置作为参数调用 MoveTo(x1, y1, x2, y2)判断并做地图更新。如果用户按空格键，则恢复游戏界面到原始地图状态，实现“重玩”功能。

```
def callback(event) :             #按键处理
    #(x1, y1)、(x2, y2)分别代表工人移动趋势方向前的两个方格
    global x,y,myArray
    print("按下键：" )
    print("按下键：", event.char)
    KeyCode=event.keysym
    #工人当前位置(x,y)
    if KeyCode=="Up":             #分析按键消息
    #向上
            x1=x;
            y1=y-1;
            x2=x;
            y2=y-2;
            #将所有位置输入以判断并做地图更新
            MoveTo(x1, y1, x2, y2);
    #向下
    elif KeyCode=="Down":
            x1=x;
            y1=y+1;
            x2=x;
            y2=y+2;
            MoveTo(x1, y1, x2, y2);
```

```
    #向左
    elif KeyCode=="Left":
            x1=x-1;
            y1=y;
            x2=x-2;
            y2=y;
            MoveTo(x1, y1, x2, y2);
    #向右
    elif KeyCode=="Right":
            x1=x+1;
            y1=y;
            x2=x+2;
            y2=y;
            MoveTo(x1, y1, x2, y2);
    elif KeyCode=="Space":                #空格键
       print("按下键：", event.char)
       myArray=copy.deepcopy(myArray1)    #恢复原始地图
       drawGameImage()
```

IsInGameArea(row, col)判断是否在游戏区域中。

```
def IsInGameArea(row, col) :
    return(row>=0 and row<7 and col>=0 and col<7)
```

MoveTo(x1,y1,x2,y2)方法是最复杂的部分，实现前面分析的所有规则和对应算法。

```
def MoveTo(x1, y1, x2, y2) :
    global x,y
    P1=None                           #P1、P2 是移动趋势方向前的两个格子
    P2=None
    if IsInGameArea(x1, y1) :         #判断是否在游戏区域
        P1=myArray[x1][y1];
    if IsInGameArea(x2, y2) :
        P2=myArray[x2][y2];
    if P1==Passageway :               #P1 处为通道
        MoveMan(x,y);
        x=x1; y=y1;
        myArray[x1][y1]=Worker;
    if P1==Destination :              #P1 处为目的地
        MoveMan(x, y);
        x=x1; y=y1;
        myArray[x1][y1]=WorkerInDest;
    if P1==Wall or not IsInGameArea(x1, y1) :
        #P1 处为墙或出界
        return;
    if P1==Box :                      #P1 处为箱子
       if P2==Wall or not IsInGameArea(x1, y1) or P2==Box:#P2 处为墙或出界
```

```
            return;
        #以下 P1 处为箱子
        #P1 处为箱子，P2 处为通道
        if P1==Box and P2==Passageway :
            MoveMan(x, y);
            x=x1; y=y1;
            myArray[x2][y2]=Box;
            myArray[x1][y1]=Worker;
        if P1==Box and P2==Destination :
            MoveMan(x, y);
            x=x1; y=y1;
            myArray[x2][y2]=RedBox;
            myArray[x1][y1]=Worker;
        #P1 处为放到目的地的箱子，P2 处为通道
        if P1==RedBox and P2==Passageway :
            MoveMan(x, y);
            x=x1; y=y1;
            myArray[x2][y2]=Box;
            myArray[x1][y1]=WorkerInDest;
        #P1 处为放到目的地的箱子，P2 处为目的地
        if P1==RedBox and P2==Destination :
            MoveMan(x, y);
            x=x1; y=y1;
            myArray[x2][y2]=RedBox;
            myArray[x1][y1]=WorkerInDest;
        drawGameImage()
        #这里要验证是否过关
        if IsFinish() :
            showinfo(title="提示",message="恭喜你顺利过关" )
            print("下一关")
```

MoveMan(x,y)移走位置为(x, y)的工人，修改格子状态值。

```
    def MoveMan(x, y) :
        if myArray[x][y]==Worker :
            myArray[x][y]=Passageway;
        elif myArray[x][y]==WorkerInDest :
            myArray[x][y]=Destination;
```

IsFinish()验证是否过关，只要方格状态存在目的地（Destination）或人在目的地上（WorkerInDest），则表明有没放好的箱子，游戏还未成功，否则成功。

```
    def IsFinish():                    #验证是否过关
        bFinish=True;
        for i in range(0,7) :          #0～6
           for j in range(0,7) :       #0～6
```

```
            if(myArray[i][j]==Destination
                  or myArray[i][j]==WorkerInDest) :
                bFinish=False;
        return bFinish;
```

❹ 主程序

```
root=Tk()
root.title(" 推箱子～夏敏捷 ")
cv=Canvas(root, bg='green', width=226, height=226)
myArray=copy.deepcopy(myArray1)
drawGameImage()
cv.bind("<KeyPress>", callback)
cv.pack()
cv.focus_set()          #将焦点设置到 cv 上
root.mainloop()
```

至此完成推箱子游戏。读者可以考虑一下多关推箱子游戏如何开发，例如把 10 关游戏的地图信息存储在 map.txt 文件里，需要时从文件中读取下一关数据即可。

第12章

娱乐游戏——两人麻将游戏

12.1 麻将游戏介绍

视频讲解

麻将起源于中国，它集益智性、趣味性、博弈性于一体，是中国传统文化的一个重要组成部分，不同地区的游戏规则稍有不同。麻将牌每副 136 张，主要有“饼（文钱）”“条（索子）”“万（万贯）”等。与其他牌形式相比，麻将的玩法最为复杂、有趣，它的基本打法简单，因此成为中国历史上最能吸引人的博戏形式之一。

12.1.1 麻将术语

麻将术语是“吃”“碰”“杠”“听”。

- 吃：如果任何一位选手手中的牌的两张加上上家选手刚打下的一张牌恰好成顺子，他就可吃牌。
- 碰：如果某方打出一张牌，而自己手中有两张以上与该牌相同的牌，可以选择碰牌。碰牌后，取得对方打出的这张牌，加上自己提供的两张相同的牌成为刻子，倒下这个刻子，不能再出。然后再出一张牌。“碰”比“吃”优先，如果要碰的牌刚好是出牌方下家要吃的牌，则吃牌失败，碰牌成功。
- 杠：其他人打出一张牌，自己手中有 3 张相同的牌，即可杠牌。杠牌分明杠和暗杠两种。
- 听：当将手中的牌都凑成了有用的牌，只需再加上第 14 张便可和牌，则玩家就可以进入听牌的阶段。

12.1.2 牌数

麻将共 136 张牌。

（1）万牌：从一万至九万，各 4 张，共 36 张。
（2）饼牌：从一饼至九饼，各 4 张，共 36 张。
（3）条牌：从一条至九条，各 4 张，共 36 张。
（4）风牌：东、南、西、北，各 4 张，共 16 张。
（5）字牌：中、发、白，各 4 张，共 12 张。

本章设计的是两人麻将程序，可以实现玩家（人）和计算机对下。游戏有吃、碰功能，和牌（即胡牌）判断。为了降低程序的复杂度，游戏没有设计杠的功能。同时对计算机出牌进行了智能设计，游戏中上方为计算机的牌，下方为玩家的牌，有“吃牌”“碰牌”“和牌”“摸牌”和“出牌”按钮供玩家选择，游戏初始界面如图 12-1 所示。

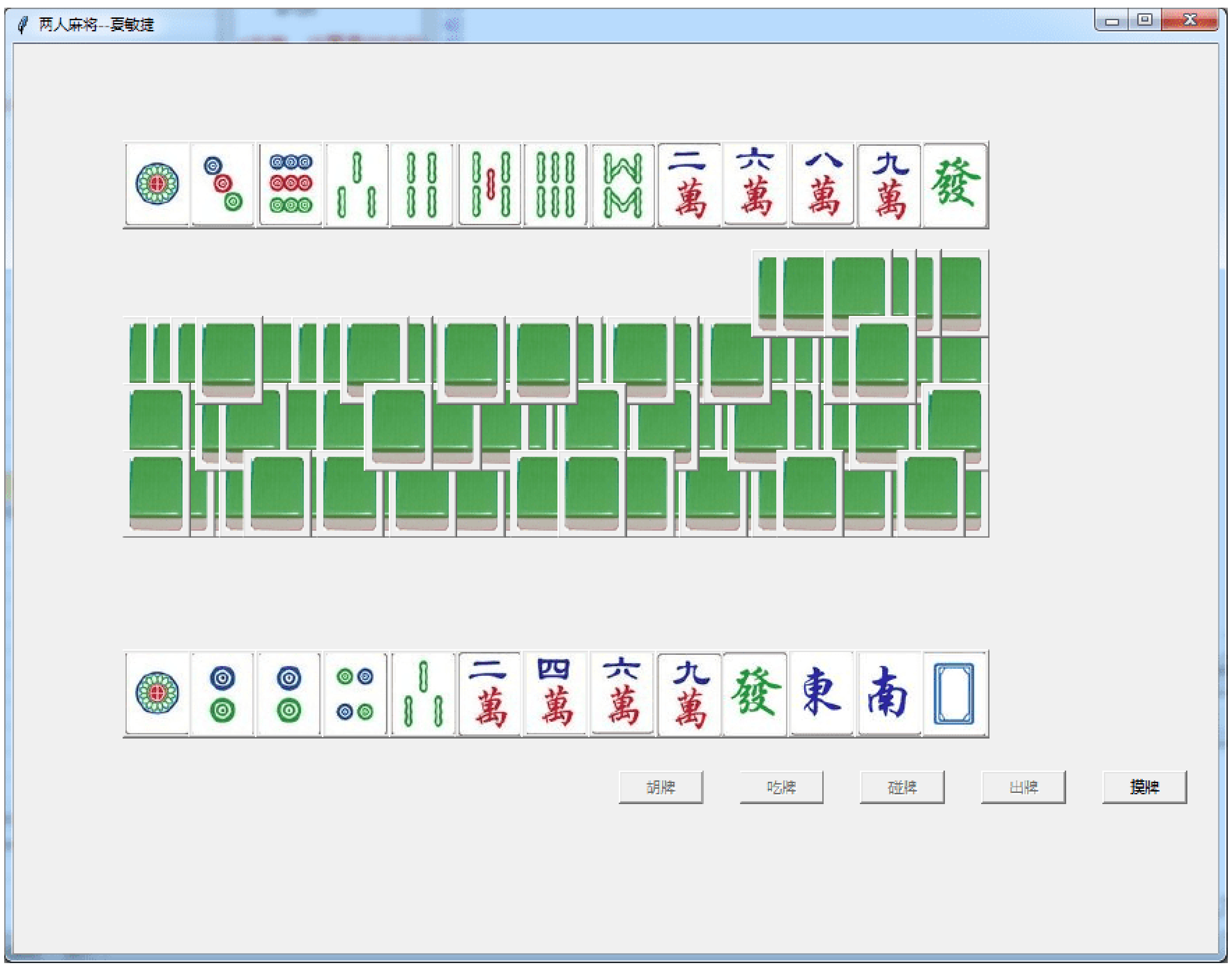


图 12-1　两人麻将游戏运行的初始界面

12.2　两人麻将游戏设计的思路

12.2.1　素材图片

麻将牌共 136 张。万子牌从一万至九万，饼子牌从一饼至九饼，条子牌从一条至九条，字牌有东、南、西、北和中、发、白。在设计时麻将牌图片文件按以下规律编号：一饼至九饼为 11.jpg～19.jpg，一条至九条为 21.jpg～29.jpg，一万至九万为 31.jpg～39.jpg，字牌为 41.jpg～47.jpg，如图 12-2 所示。

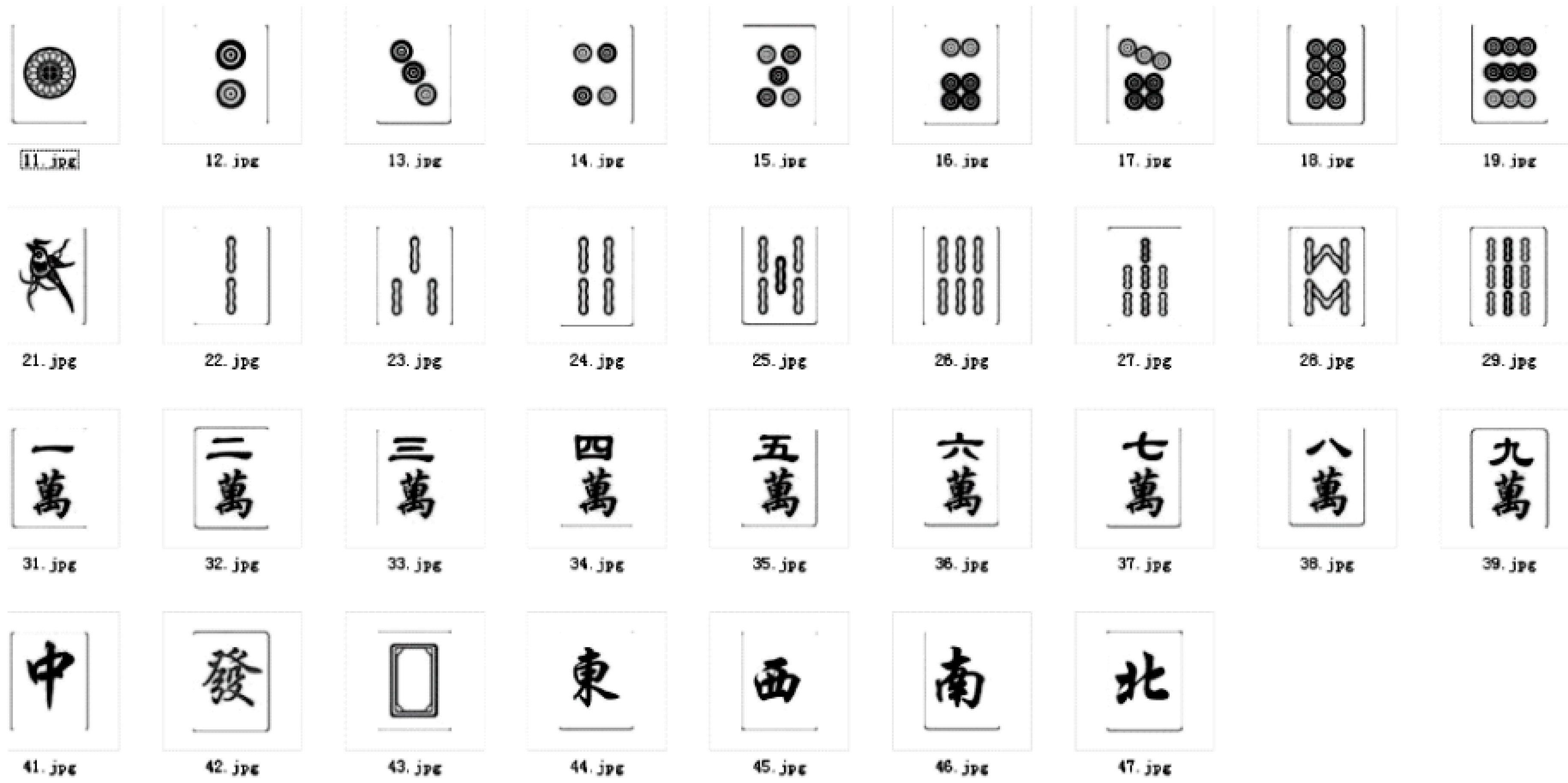


图 12-2　素材图片

12.2.2　游戏的逻辑实现

视频讲解

玩家自己出过牌，MyTurn=False，则轮到计算机智能出牌，计算机出完牌则 MyTurn= True，同时“摸牌”按钮有效，这样又轮到玩家出牌。

```
MyTurn=True                #轮到玩家出牌
Get_btn["state"]=NORMAL       #摸牌有效
```

在游戏过程中， playersCard 列表（数组）记录两个牌手的牌，其中 playersCard[0]记录玩家自己（0 号牌手）的牌，playersCard[1]记录计算机（1 号牌手）的牌。同理，playersOutCard 数组记录两个牌手出过的牌。所有的牌存入 m_aCards 列表（数组），同时为了便于知道该发哪张牌，这里 k 记录已发牌的个数，从而知道要摸的牌是 m_aCards[k]。

12.2.3　碰/吃牌的判断

在游戏过程中玩家自己可以碰牌和吃牌，所以需要判断计算机（1 号牌手）刚出的牌玩家是否可以碰、吃，如果能够碰、吃，则“碰牌”“吃牌”和“摸牌”按钮有效。

能否碰牌的判断比较简单，由于每张牌对应文件的主文件名是 imageID，所以仅仅统计相同 imageID 的牌即可知道是否有两张以上，如果有则可以碰牌。

```
#是否可以碰牌
def canPeng(a,card):#(List a,Card card)
    n=0
    for i in range(0,len(a)):
        c=a[i]
        if(c.imageID==card.imageID):
            n+=1
    if n>=2:
```

```
        return True
    print("不能碰牌!!!",card.imageID)
    return False
```

能否吃牌的判断也比较简单，由于牌手手里的牌（a 列表）已经排过序了，只要判断以下 3 种情况：

1**

1

**1

1 代表对方刚出的牌，如果符合这 3 种情况则可以吃牌。

```
#是否可以吃牌
def canChi(a,card):
    n=0
    if card.m_nType==4:          #字牌不用判断吃
        return False
    for i in range(0,len(a)-1): #1**
        c1=a[i]
        c2=a[i+1]
        if(c1.m nNum==card.m nNum+1 and c1.m nType==card.m nType
          and c2.m nNum==card.m nNum+2 and c2.m nType==card.m nType):
            return True
    for i in range(0,len(a)-1): #*1*
        c1=a[i]
        c2=a[i+1]
        if(c1.m nNum==card.m nNum-1 and c1.m nType==card.m nType
          and c2.m nNum==card.m nNum+1 and c2.m nType==card.m nType):
            return True
    for i in range(0,len(a)-1): #**1
        c1=a[i]
        c2=a[i+1]
        if(c1.m nNum==card.m nNum-2 and c1.m nType==card.m nType
          and c2.m nNum==card.m nNum-1 and c2.m nType==card.m nType):
            return True
    print("不能吃牌!!!",card.imageID)
    return False
```

12.2.4　和牌算法

视频讲解

❶ 数据结构的定义

麻将由“万”“饼”（筒）“条”（索）“字” 4 类牌组成，其中“万”又分为“一万”～“九万”各 4 张，共 36 张，“饼”和“条”类似，“字”又分为“东”“南”“西”“北”“中”“发”“白”各 4 张，共 28 张。

这里定义了一个 4×10 的二维列表（相当于其他语言中的 4×10 的二维数组 int allPai [4][10]），它记录了牌手手中的牌的全部信息，行号记录类别信息，第 0～3 行分别代表“饼”“条”“万”“字”。

以第 2 行为例，它的第 0 列记录了牌中所有“万”的总数，第 1～9 列分别对应“一万”～“九万”的个数，“饼”和“条”类似。“字”不同的是第 1～7 列对应的是“中”“发”“白”“东”“南”“西”“北”的个数，第 8、9 列恒为 0。

根据麻将的规则，数组中牌的总数一定为 3n+2，其中 n=0,1,2,3,4。例如有以下数组：

```
allPai=[
        [6,1,1,1,0,3],    #饼，6 个饼牌，“一饼”“二饼”“三饼”和 3 个“五饼”
        [5,0,2,0,3],      #条，5 个条牌，两个“二条”和 3 个“四条”
        [0],              #万，无万牌
        [3,0,3]           #字，3 个字牌“发”
        ]
```

它表示牌手手中的牌为“一饼”“二饼”“三饼”“五饼”“五饼”“五饼”，“二条”“二条”“四条”“四条”“四条”，“发”“发”“发”，共 6 张“饼”、5 张“条”、0 张“万”、3 张“字”。

❷ 算法设计

由于“七对子”“十三幺”这种特殊牌型的和牌依据不是牌的相互组合，而且规则也不尽相同，这里将这类情况排除在外。

尽管能构成和牌的形式千变万化，但稍加分析可以看出它离不开一个模型：可以分解为“三、三……三、二”的形式（总牌数为 3n+2 张），其中的“三”表示的是“顺”或“刻”（连续 3 张牌叫作“顺”，例如“三饼”“四饼”“五饼”，“字”牌不存在“顺”；3 张同样的牌叫作“刻”，例如“三饼”“三饼”“三饼”）；其中的“二”表示的是“将”（两张相同的牌可作为“将”，例如“三饼”“三饼”）。

在代码实现中，首先判断牌手手中的牌是否符合这个模型，这样就用极少的代价排除了大多数情况，具体方法是用 3 除 allPai [i][0]（存储每种牌型数量），其中 i=0, 1, 2, 3，只有在余数有且仅有一个为 2，其余全为 0 的情况下才可能构成和牌。

对于余数为 0 的牌，它一定要能分解成一个“刻”和“顺”的组合，这是一个递归的过程，由函数 bool Analyze(list,bool)处理。

对于余数为 2 的牌，一定要能分解成一对“将”与“刻”和“顺”的组合，由于任何数目大于等于 2 的牌均有作为“将”的可能，需要对每张牌进行轮询，如果它的数目大于等于 2，去掉这对“将”后再分析它能否分解为“刻”和“顺”的组合，这个过程的开销相对较大，放在了程序的最后进行处理。在递归和轮询过程中，尽管每次去掉了某些牌，但最终都会再次将这些牌加上，使得数组中的数据保持不变。

最后分析递归函数 bool Analyze(list,bool)，列表（数组）参数表示一类牌，即“万”“饼”“条”“字”之一，布尔参数指出列表（数组）参数是否为“字”牌，这是因为“字”牌只能“刻”不能“顺”。对于列表（数组）中的第 1 张牌，如果要构成和牌，它就必须与其他牌构成“顺”或“刻”。

如果数目大于等于 3，那么它们一定是以“刻”的形式组合。例如当前有 3 张“五万”，如果它们不构成“刻”，则必须有 3 张“六万”、3 张“七万”与其构成 3 个“顺”（注意此时“五万”是数组中的第 1 张牌），否则就会剩下“五万”不能组合，而此时的 3 个“顺”实际上也是 3 个“刻”。去掉这 3 张牌，递归调用 bool Analyze(list,bool)函数，如果成功则和牌。当该牌不是字牌且其下两张牌均存在时它还可以构成“顺”，去掉这 3 张牌，递归调用 bool Analyze (list,bool)函数，如果成功则和牌。如果此时还不能构成和牌，说明该牌不能与其他牌顺利组合，传入的参数不能分解为“顺”和“刻”的组合，不可以构成和牌。

这里根据上述思想单独设计一个类文件（huMain.py）验证和牌算法，代码如下：

```
class huMain():
    def __init__(self):                              #构造函数
        #定义牌手手中的牌 int allPai[4][10]
        self.allPai=[[6,1,4,1,0,0,0,0,0,0], #饼
                     [3,1,1,1,0,0,0,0,0,0],      #条
                     [0,0,0,0,0,0,0,0,0,0],      #万
                     [5,2,3,0,0,0,0,0,0,0]]      #字
        if self.Win(self.allPai):
            print("Hu!\n")
        else:
            print("Not Hu!\n")
    #判断是否和牌的函数
    def Win(self,allPai):
        jiangPos=0                                   #“将”的位置
        jiangExisted=False
        #是否满足 3,3,3,3,2 模型
        for i in range(0,4):
            #yuShu                                    #余数
            yuShu=allPai[i][0]%3
            if  yuShu==1 :
                return False                         #不满足 3，3，3，3，2 模型
            if  yuShu==2 :
                if  jiangExisted==True:
                    return False                     #不满足 3，3，3，3，2 模型
                jiangPos=i                           #“将”在哪行
                jiangExisted=True

        #不含“将”处理
        for i in range(0,4):
            if  i!=jiangPos :
                if  not self.Analyze(allPai[i],i==3):
                    return False

        #该类牌中要包含“将”，因为要对“将”进行轮询，效率较低，放在最后
```

```
    success=False                    #指出除掉“将”后能否通过
    for j in range(1,10):            #对列进行操作，用 j 表示
        if(allPai[jiangPos][j]>=2):
            #除去这两张将牌
            allPai[jiangPos][j]-=2
            allPai[jiangPos][0]-=2
            if self.Analyze(allPai[jiangPos],jiangPos==3) :
                    success=True
            #还原这两张将牌
            allPai[jiangPos][j]+=2
            allPai[jiangPos][0]+=2
            if success==True :
                break
    return success

#分解成“刻”“顺”组合
def Analyze(self,aKindPai,ziPai):   #(int aKindPai[],Boolean ziPai)
    if aKindPai[0]==0 :
        return True
    #寻找第 1 张牌
    for j in range(1,10):
        if aKindPai[j]!=0:
          break
    if aKindPai[j]>=3:               #作为刻牌
        #除去这 3 张刻牌
        aKindPai[j]-=3
        aKindPai[0]-=3
        result=self.Analyze(aKindPai,ziPai)
        #还原这 3 张刻牌
        aKindPai[j]+=3
        aKindPai[0]+=3
        return result
    #作为顺牌
    if(not ziPai)and(j<8) and(aKindPai[j+1]>0) and(aKindPai[j+2]>0):
        #除去这 3 张顺牌
        aKindPai[j]-=1
        aKindPai[j+1]-=1
        aKindPai[j+2]-=1
        aKindPai[0]-=3
        result=self.Analyze(aKindPai,ziPai)
        #还原这 3 张顺牌
        aKindPai[j]+=1
        aKindPai[j+1]+=1
        aKindPai[j+2]+=1
        aKindPai[0]+=3
```

```
            return result
        return False
```

12.2.5　实现计算机智能出牌

在游戏中有两个牌手，一个是玩家自己（0 号牌手），一个是计算机（1 号牌手）。如果计算机只能随机出牌，则游戏的可玩性较差，所以智能出牌是一个设计重点。

为了判断出牌，需要首先计算牌手手中各种牌型的数量。二维列表 paiArray 存储了和牌算法的数据结构，记录了牌手手中牌的全部信息，行号记录类别信息，第 0～3 行分别代表“饼”“索”“万”“字”。本游戏这里给出一个智能出牌的算法：

假设 cards 为手中所有的牌。

（1）判断字牌的单张，即 paiArray 行号为 3 的元素是否为 1，有则找到，返回在 cards 的索引号。

（2）判断顺子、刻子（3 张相同的），有则从 paiArray 中消去，即不需要考虑这些牌。

（3）判断单张非字牌（饼、条、万），有则找到，返回在 cards 中的索引号。

（4）判断两张牌（饼、条、万，包括字牌），有则找到（即拆双牌），返回在 cards 中的索引号。

（5）如果以上情况均没出现，则随机选出 1 张牌，当然此种情况一般不会出现。

```
#计算机智能出牌 v1.0，计算出牌的索引号
def  ComputerCard(cards):
    #计算牌手手中各种牌型的数量
    paiArray=[[0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0],
              [0,0,0,0,0,0,0,0,0,0]]
    for i in range(0,14):
        card=cards[i]
        if(card.imageID>10 and card.imageID<20):    #饼
                paiArray[0][0]+=1
                paiArray[0][card.imageID-10]+=1
        if(card.imageID>20 and card.imageID<30):    #条
                paiArray[1][0]+=1
                paiArray[1][card.imageID-20]+=1
        if(card.imageID>30 and card.imageID<40):    #万
                paiArray[2][0]+=1
                paiArray[2][card.imageID-30]+=1
        if(card.imageID>40 and card.imageID<50):    #字
                paiArray[3][0]+=1
                paiArray[3][card.imageID-40]+=1
    print(paiArray)
    #计算机智能选牌
    #（1）判断字牌的单张，有则找到
```

```
    for j in range(1,10):
        if(paiArray[3][j]==1):
            #获取在手中的牌的位置下标
            k=ComputerSelectCard(cards,3+1,j)
            return k
    #（2）判断顺子、刻子（3 张相同的）
    for i in range(0,3):
        for j in range(1,10):
            if(paiArray[i][j]>=3):        #刻子
                    paiArray[i][j]-=3
            if(j<=7 and paiArray[i][j]>=1 and  paiArray[i][j+1]>=1
               and  paiArray[i][j+2]>=1): #顺子
                    paiArray[i][j]-=1
                    paiArray[i][j+1]-=1
                    paiArray[i][j+2]-=1
    #（3）判断单张非字牌（饼、条、万），有则找到
    for i in range(0,3):
        for j in range(1,10):
            if(paiArray[i][j]==1):
                #获取在手中的牌的位置下标
                k=ComputerSelectCard(cards,i+1,j)
                return k
    #（4）判断两张牌（饼、条、万，包括字牌），有则找到，拆双牌
    for i in range(3,-1):
        for j in range(1,10):
            if(paiArray[i][j]==2):
                #获取在手中的牌的位置下标
                k=ComputerSelectCard(cards,i+1,j)
                return k
    #（5）如果以上情况均没出现，则随机选出 1 张牌
    k=random.randint(0,13)              #随机选出 1 张牌
    return k
#根据牌（花色 nType，点数 nNum）找在 a 数组中的索引位置
def ComputerSelectCard(a, nType,nNum):
    for i in range(0,len(a)):
        card=a[i]
        if(card.m nType==nType and card.m nNum==nNum):
                return i
    return -1
```

12.3　关键技术

12.3.1　声音的播放

winsound 模块可以访问由 Windows 平台提供的基本的声音播放设备，它包含数个声音播放函数和常量。

❶ **Beep(frequency, duration) 函数**

计算机蜂鸣器。其中，frequency 参数指定声音的频率，单位为赫兹，并且必须在 37～32767 的范围之中；duration 参数指定声音应该持续的毫秒数。

❷ **PlaySound(sound, flags)函数**

从 Windows 平台 API 中调用 PlaySound()函数。其中，sound 参数必须是一个由文件名、音频数据形成的字符串，或为 None。它的解释依赖于 flags 的值，该值可以是一个位方式或下面变量的组合。

- SND_FILENAME：sound 参数是一个 WAV 文件的文件名。
- SND_LOOP：重复地播放声音。
- SND_MEMORY：提供给 PlaySound()的 sound 参数是一个 WAV 文件的内存映像形成的字符串。
- SND_PURGE：停止播放所有指定声音的实例。
- SND_ASYNC：立即返回，允许声音异步播放。
- SND_NOSTOP：不中断当前播放的声音。
- MB_ICONASTERISK：播放 SystemDefault 声音。
- MB_ICONEXCLAMATION：播放 SystemExclamation 声音。

例如播放八柄.wav 声音文件的代码如下：

```
import winsound
winsound.PlaySound("res\\sound\\八柄.wav", winsound.SND_FILENAME)
```

12.3.2　返回对应位置的组件

在 Python Tkinter 中鼠标单击某组件，如何得到对应位置的组件呢?

实际上，当鼠标单击，参数 event 的 event.x 和 event.y 可以获取鼠标坐标的时候，event.widget 返回的就是事件发生时所在的组件，也就是被用户单击的组件。

例如当用户选麻将牌时，系统自动调用鼠标按下事件函数，其中将被单击的麻将牌上移 20 像素。如果此麻将牌已被选过，则下移 20 像素恢复到原来的正常位置。

```
def btn_MouseDown(event):     #鼠标单击按下事件函数
    #找到相应的麻将牌对象
    card=event.widget         #event.widget 获取触发事件的对象
    card.y-=20                #上移 20 像素
    card.place(x=event.widget.x,y=event.widget.y)
```

```
    if(m_LastCard==None):          #未选过的牌
        m_LastCard=card
        PlayerSelectCard=card
    else:                          #已经选过的牌
        m_LastCard.MoveTo(m_LastCard.getX(),m_LastCard.getY()+20)#下移20像素
        m_LastCard=card
        PlayerSelectCard=card
```

12.3.3 对保存麻将牌的列表排序

Python 语言中的列表排序方法有 3 个，即 reverse()（反转/倒序排序）、sort()（正序排序）、sorted()（获取排序后的列表），后两种方法还可以加入条件参数进行排序。

❶ reverse()方法

将列表中的元素倒序，把原列表中的元素顺序从右至左重新存放。例如下面这样：

```
>>> x=[1,5,2,3,4]
>>> x.reverse()
>>> x                   #结果是[4, 3, 2, 5, 1]
```

❷ sort()方法

此方法对列表内容进行正向排序，排序后的新列表会覆盖原列表（ID 不变），是就地排序，以节约空间。也就是说，sort()排序方法是直接修改原列表。

```
>>> a=[5,7,6,3,4,1,2]
>>> a.sort()
>>> a                   #结果是[1, 2, 3, 4, 5, 6, 7]
```

❸ sorted()方法

该方法既可以保留原列表，又能得到已经排好序的列表，其操作方法如下：

```
>>> a=[5,7,6,3,4,1,2]
>>> b=sorted(a)
>>> a                   #结果是[5, 7, 6, 3, 4, 1, 2]
>>> b                   #结果是[1, 2, 3, 4, 5, 6, 7]
```

注意：使用 sort()方法和 sorted()方法可以加入参数。

列表的元素可以是各种类型，例如字符串、字典、用户自己定义的类。如果不使用内置比较函数，可以使用参数：

```
sort(cmp=None, key=None, reverse=False)
sorted(cmp=None, key=None, reverse=False)
```

其中，cmp 和 key 都是函数，这两个函数作用于列表的元素上产生一个结果，sorted()方法根据这个结果来排序。reverse 是一个布尔值，表示是否反转比较结果。

cmp(e1,e2)是带两个参数的比较函数，当返回值为负数时 e1<e2，为 0 时 e1==e2，为正数时 e1>e2，默认为 None，即用内置的比较函数。例如：

```
>>>students=[('张海',20),('李斯',19),('赵大强',31),('王磊',14)]
>>>students.sort(cmp=lambda x,y:cmp(x[1],y[1]))   #按年龄数字大小排序
>>>students
```

结果如下：

```
[('王磊', 14), ('李斯', 19), ('张海', 20), ('赵大强', 31)]
```

key 是带一个参数的函数，用来为每个元素提取比较值。其默认为 None，即直接比较每个元素。通常，key 比 cmp 快很多，因为对每个元素，key 只处理一次，而 cmp 会处理多次。例如：

```
>>>students=[('张海',20),('李斯',19),('赵大强',31),('王磊',14)]
>>>students.sort(key=lambda x:x[1])
>>>students
```

结果如下：

```
[('王磊', 14), ('李斯', 19), ('张海', 20), ('赵大强', 31)]
```

用元素已经命名的属性做 key：

```
students.sort(key=lambda student: student.age)
```

用 operator 函数来加快速度，上面的排序等价于：

```
>>> from operator import itemgetter, attrgetter
>>> students.sort(key=itemgetter(2))
>>> students.sort(key=attrgetter('age'))
```

说明：cmp 参数在 Python3.0 以后不再支持，所以 Python3.5 只能使用 key、reverse 参数。

在本章中需要按花色理牌手手中的牌，使用的就是 sort()排序，参数 key 使用的是麻将牌的图像 ID 属性。由于麻将牌的图像 ID 是有次序的，从而实现按花色理牌。

```
def sortPoker2(cards):             #按花色理牌手手中的牌
    n=len(cards)                   #元素（牌）的个数
    cards.sort(key=operator.attrgetter('imageID'))#按麻将牌的图像 ID 属性排序
    print("排序后")
```

12.4　两人麻将游戏设计的步骤

12.4.1　设计麻将牌类

Card.as 为麻将牌类（继承按钮组件 Button），构造函数根据参数 type 指定麻将牌的类型，参数 num 指定麻将牌的点数。从牌的类型和牌的点数计算出对应的麻将牌图片。麻将牌的所有图片见图 12-2 所示的素材。

视频讲解

Card 麻将牌类可以实现麻将牌正面、背面的显示以及移动的功能。

```
#Card 麻将牌类
'''m_bFront 表示是否显示牌正面的标志
   m_nType 表示牌的类型 饼=1 条=2 万=3 字牌=4
   m_nNum 表示牌的点数（1～9）
   FrontURL 表示牌文件的 URL 路径
   imageID 表示牌自己的图像编号 ID
   cardID 表示牌自己在数组中的索引 ID
   x,y 表示牌的坐标
'''
#可以实现麻将牌正面、背面的显示以及移动的功能
class Card(Button):
    #构造函数，参数 type 指定牌的类型，参数 num 指定牌的点数
    def __init__(self,cardtype,num,bm,master):
        Button.__init__(self,master)
        self.m nType=cardtype          #牌的类型 饼=1 条=2 万=3 字牌=4
        self.m nNum=num                #牌的点数（1～9）
        #根据牌的类型及编号来设置牌文件的路径及文件名
        if self.m nType==1 :           #桶（饼）
            FrontURL="res/nan/1"
        elif self.m_nType== 2 :        #条
            FrontURL="res/nan/2"
        elif self.m_nType== 3 :        #万
            FrontURL="res/nan/3"
        elif self.m nType== 4 :        #字牌
            FrontURL="res/nan/4"
        self.img=bm
        self.imageID=self.m_nType * 10 + self.m_nNum    #牌自己的图像编号 ID
        FrontURL=FrontURL + str(self.m_nNum)            #URL 地址
        FrontURL=FrontURL + ".png"
        self["width"]=51               #麻将牌方块的宽度
        self["height"]=67              #麻将牌方块的高度
        self["text"]=str(self.imageID)+".png"
        self.setFront(False)
        #self.MoveTo(100, 100)
        self.bind("<ButtonPress>",btn MouseDown)
        self.cardID=0
    def __cmp__(self, other):
        return cmp(self.imageID, other.imageID)
    def setFront(self, b):             #是否显示牌正面
        self.m_bFront=b
```

```
        if(b==True):
            self["image"]=self.img      #显示牌正面图片
        else:
            self["image"]=back          #显示牌背面图片
    def MoveTo(self, x1, y1):           #移到指定的(x1, y1)位置
        self.place(x=x1, y=y1)
        self.x=x1                       #牌的坐标
        self.y=y1
    def getX(self):
        return self.x
    def getY(self):
        return self.y
    def getImageID(self):               #牌自己的图像编号 ID
        return imageID
#------------------------------------Card end
```

12.4.2　设计游戏主程序

视频讲解

导入包及相关的类：

```
from tkinter import *
import random
from threading import Timer
import time
import operator
import winsound   #声音模块
from tkinter.messagebox import *
```

创建窗口对象，imgs 用来存储麻将图片。

```
win=Tk()                                    #创建窗口对象
win.title("两人麻将——夏敏捷")                 #设置窗口标题
win.geometry("995x750")
imgs=[]                                     #存储麻将的正面图片
back=PhotoImage(file='res\\bei.png')        #存储牌的背面图片
m aCards=[]                                 #存储 136 张麻将牌的列表
playersCard=[[],[]]                         #记录两个牌手拿到的牌
playersOutCard=[[],[]]                      #记录两个牌手出过的牌
k=0                                         #记录已发出牌的个数
m LastCard=None                             #用户是否选过牌
PlayerSelectCard=None                       #用户选中的牌
MyTurn=True                                 #轮到玩家出牌（游戏开始玩家先出牌）
```

实例化“吃牌”“碰牌”“和牌”“摸牌”按钮，由于还未发牌，所以这些按钮均设置

为无效。

```
#功能按钮
Get_btn=Button(win,text="摸牌", command=OnBtnGet_Click)
Peng_btn=Button(win,text="碰牌",command=OnBtnChi_Click)
Chi_btn=Button(win,text="吃牌", command=OnBtnChi_Click)
Out_btn=Button(win,text="出牌", command=OnBtnOut_Click)
Win_btn=Button(win,text="和牌", width=70,height=27)
Win_btn.place(x=500,y=600,width=70,height=27)
Chi_btn.place(x=600,y=600,width=70,height=27)
Peng_btn.place(x=700,y=600,width=70,height=27)
Out_btn.place(x=800,y=600,width=70,height=27)
Get_btn.place(x=900,y=600,width=70,height=27)
#Get_btn.pack_forget()                  #隐藏 button
#Get_btn["state"]=DISABLED              #“摸牌”按钮无效
Peng_btn["state"]=DISABLED              #“碰牌”按钮无效
Chi_btn["state"]=DISABLED               #“吃牌”按钮无效
Out_btn["state"]=DISABLED               #“出牌”按钮无效
Win_btn["state"]=DISABLED               #“和牌”按钮无效
BeginGame()                             #开始游戏，玩家先出牌
win.mainloop()
```

BeginGame()函数加载 136 张麻将牌到舞台，同时重置游戏，完成洗牌功能，即随机交换 m_aCards 中的两张牌；并将 136 张麻将牌的背面显示在舞台上，设置两家 26 张初始麻将牌的位置。

```
def BeginGame():                        #开始游戏，玩家先出牌
    MyTurn=True
    LoadCards()                         #加载 136 张麻将牌到舞台
    random.shuffle(m_aCards)            #洗牌操作，将列表中的元素打乱
    ResetGame()                         #初始发 26 张牌给玩家和计算机
```

LoadCards()创建 136 张麻将牌，并将牌添加到游戏舞台和 m_aCards 列表（数组）中。

```
def LoadCards():                        #加载 136 张麻将牌到舞台
    for m_nType in range(1,4):          #1～3 代表饼、条、万
        for num in range(1,10):         #1～9
            #根据牌的类型及编号来设置牌文件的路径及文件名
            if m_nType==1:              #桶（饼）
                FrontURL="res/nan/1"
            elif m_nType==2 :       #条
                FrontURL="res/nan/2"
            elif m_nType==3 :       #万
                FrontURL="res/nan/3"

            FrontURL=FrontURL+str(num)     #URL 地址
```

```
            FrontURL=FrontURL+".png"
            imgs.append(PhotoImage(file=FrontURL))
            for n in range(1,5):                #1～4，每种牌 4 张
                card=Card(m_nType,num,imgs[len(imgs)-1],win)#创建“饼、条、万”牌
                #card.MoveTo(100+num*60,100+m_nType*80)
                m_aCards.append(card)           #将牌添加到列表（数组）

    cardtype=4                                  #字牌
    for num in range(1,8):                      #1～7，7 种字牌
        FrontURL="res/nan/4"
        FrontURL=FrontURL+str(num)          #URL 地址
        FrontURL=FrontURL+".png"
        imgs.append(PhotoImage(file=FrontURL))
        for n in range(1,5):                    #每种牌 4 张
            card=Card(cardtype,num,imgs[len(imgs)-1],win)     #创建字牌
            #card.MoveTo(100+num*60,100+4*80)
            #card["state"]=DISABLED
            m_aCards.append(card)               #将牌添加到列表（数组）
```

ResetGame()在洗牌操作后将 136 张麻将牌的背面显示在舞台上，并完成发牌功能，共发给两个玩家 26 张麻将牌，同时设置 26 张初始麻将牌的位置。

```
def ResetGame():                                #发给两家 26 张麻将牌
    playersCard[0]=[]                           #玩家手中的牌
    playersCard[1]=[]                           #计算机的牌
    for n in range(0,len(m_aCards)):            #重新设置 136 牌在场景中的位置
        m_aCards[n].x=90+20*(n%34)
        m_aCards[n].y=170+55*(n-n%34)/34
        m_aCards[n].MoveTo(m_aCards[n].x, m_aCards[n].y)
        #m_aCards[n].setComponentZOrder(m_aCards[n], n)
        m_aCards[n].setFront(False)             #显示麻将牌的背面
    #开始发牌
    ShiftCards()
    m_LastCard=None                             #上次用户所选择的卡片
    playersOutCard[0]=[]                        #玩家出过的牌
    playersOutCard[1]=[]                        #计算机出过的牌
```

ShiftCards()发给两个玩家 26 张麻将牌，每个玩家发完 13 张牌以后，需要调用sortPoker2(cards)按花色理牌手手中的牌。

```
def ShiftCards():
    global k
    for k in range(0,26):                   #发牌，设置最初发的 26 张麻将牌的位置
         Shift(k)
    print("玩家按花色理手中的牌")
    sortPoker2(playersCard[0])              #玩家按花色理手中的牌
    print("计算机按花色理手中的牌")
```

```
    sortPoker2(playersCard[1])             #计算机按花色理手中的牌
    OuterPlayerNum=0                       #出牌人数为 0
    k=26                                   #发牌数量
```

Shift()设置最初 26 张麻将牌的位置，并且给发给玩家的麻将牌加上"<ButtonPress>"事件监听，当单击麻将牌时系统将调用 btn_MouseDown()事件函数，对发给玩家的对家（计算机）的麻将牌则不需要监听。

```
def Shift(k):   #设置每张麻将牌的位置
    #global k
    #print('running',k)
    i=k%2
    j=(k-k%2)/2
    if i==0 :                              #玩家自己
        m_aCards[k].setFront(True)         #显示麻将牌的正面
        m_aCards[k].MoveTo(80+55*j, 500)
        #监听每张麻将牌，当单击麻将牌时系统将调用 btn_MouseDown()
        m_aCards[k].bind("<ButtonPress>",btn_MouseDown)
    elif i==1 :                            #玩家的对家（计算机）
        m_aCards[k].MoveTo(80+55 * j, 80)
        m_aCards[k].setFront(True)         #显示麻将牌的正面
    playersCard[(k%2)].append(m_aCards[k]) #按顺序存储到记录两个牌手的牌的数组
```

sortPoker2(cards)按花色理玩家手中的牌。由于 imageID 是按照花色编号的，所以按照 imageID 大小排序就可以了。

```
def sortPoker2(cards):                     #按花色理牌手手中的牌
    n=len(cards)                           #元素（牌）的个数
    #排序
    cards.sort(key=operator.attrgetter('imageID'))
    print("排序后")
    for index in range(0,n):               #重新设置各张牌在场景中的位置
        print(cards[index].imageID)
        newx=90+55*index
        y=cards[index].getY()
        cards[index].MoveTo(newx, y)
        cards[index].cardID=index
```

玩家手中的牌可以响应鼠标单击，当用户选麻将牌时系统将调用 btn_MouseDown()事件函数。另外，通过 event.widget 可以获取用户选的麻将牌对象，将此牌上移 20 像素。如果已经选过牌，则还需要将已经选过的牌下移 20 像素。

```
#当用户选麻将牌时系统将自动调用此函数
def btn_MouseDown(event):                  #鼠标单击按下事件函数
    global m_LastCard,PlayerSelectCard
    if event.widget["state"]==DISABLED:
       return
```

```
        if(event.widget.m_bFront==False):
            return
        #找到相应的麻将牌对象
        card=event.widget                    #event.widget 获取触发事件的对象
        card.y-=20
        card.place(x=event.widget.x,y=event.widget.y)
        if(m_LastCard==None):                #未选过的牌
            m_LastCard=card
            PlayerSelectCard=card
        else:                                #已经选过的牌
            m_LastCard.MoveTo(m_LastCard.getX(),m_LastCard.getY()+20)#下移 20 像素
            m_LastCard=card
            PlayerSelectCard=card
```

以下是 4 个按钮的单击事件处理。

在“摸牌”按钮单击事件中，将 m_aCards[k]牌移动到玩家牌所在的位置，并按花色排序理牌；调用 ComputerCardNum(playersCard[0])计算玩家手中各种牌型的数量并判断是否和牌，如果和牌则游戏结束。

```
    def OnBtnGet_Click():                        #“摸牌”按钮事件
        global k
        global playersCard,MyTurn
        #玩家按花色理手中的牌
        m_aCards[k].MoveTo(90+55*13, 500)
        m_aCards[k].setFront(True)               #显示麻将牌的正面
        print("玩家手中牌 1111",len(playersCard[0]))
        playersCard[0].append(m_aCards[k])   #第 14 张牌
        #监听第 14 张牌
        m_aCards[k].bind("<ButtonPress>",btn_MouseDown)
        print("玩家手中牌 2222",len(playersCard[0]))
        sortPoker2(playersCard[0])               #按顺序存储到记录牌手的牌的数组
        result1=ComputerCardNum(playersCard[0])#计算牌手手中各种牌型的数量，判断是否和牌
        if(result1):                             #和牌了
            Win_btn["state"]=NORMAL
            showinfo(title="恭喜",message="玩家 Win!")
            return                               #玩家不需要再出牌
        k=k+1                                    #下一张要摸的牌在 m_aCards 中的索引号
        Out_btn["state"]=NORMAL                  #“出牌”按钮有效
        Chi_btn["state"]=DISABLED                #“吃牌”按钮无效
        Peng_btn["state"]=DISABLED               #“碰牌”按钮无效
        Get_btn["state"]=DISABLED                #“摸牌”按钮无效
        MyTurn=True
```

在“出牌”按钮单击事件中，将被选中的牌 PlayerSelectCard 移到左侧，并从 playersCard[0] 中删除被选中的牌 PlayerSelectCard，轮到计算机出牌时，ComputerOut()实现计算机智能出牌。

```
def OnBtnOut_Click():
    global MyTurn
    global PlayerSelectCard,m_LastCard,MyTurn
    print("出牌")
    if(MyTurn==False):                          #没轮到自己出牌
        return
    if(PlayerSelectCard==None):                 #还没选择出的牌
        showinfo(title="提示",message="还没选择出的牌")
        return
    print(PlayerSelectCard)
    if not(PlayerSelectCard==None):
        Out_btn["state"]=DISABLED               #“出牌”按钮无效
        playersOutCard[0].append(PlayerSelectCard);
        PlayerSelectCard.x=len(playersOutCard[0])*25-25;  #移动被选中的牌
        PlayerSelectCard.y=420;
        PlayerSelectCard.MoveTo(PlayerSelectCard.x, PlayerSelectCard.y);
        #outCardOrder(playersOutCard[0]);       #整理玩家出的牌的 Z 轴深度
        #玩家牌减少
        print(PlayerSelectCard.cardID)
        del(playersCard[0][PlayerSelectCard.cardID])
        #playersCard[0].remove(PlayerSelectCard);
        m_LastCard=None
        PlayerSelectCard=None
        MyTurn=False
        Out_btn["state"]=DISABLED
        ComputerOut()                           #计算机智能出牌
        fun2()                                  #游戏顺序逻辑控制
```

对于碰、吃牌，这里不再区分处理，仅仅将对家的牌加入玩家自己的 playersCard[0]列表（数组）中，对 playersCard[0]记录的牌进行排序，从而达到理牌目的。最后计算牌手手中各种牌型的数量，判断是否和牌，如果和牌则“出牌”按钮无效，否则“出牌”按钮出现，玩家选择牌后可以出牌。

```
#对于碰、吃牌，这里不再区分处理
def OnBtnChi_Click():                   #“吃牌”按钮单击事件
    global MyTurn
    card=playersOutCard[1][len(playersOutCard[1])-1];
    card.MoveTo(90+55*13, 500);
    card.setFront(True);                #显示麻将牌的正面
    playersCard[0].append(card);        #第 14 张牌
    #监听第 14 张牌
    #card.bind("<ButtonPress>",btn_MouseDown)  #不绑定事件，可以防止此牌被玩家再次出
    print("碰吃的牌是",card.imageID)
    sortPoker2(playersCard[0]);         #按顺序存储到记录玩家的牌的列表（数组）中
    result1=ComputerCardNum(playersCard[0]);#计算牌手手中各种牌型的数量，判断是否和牌
    if(result1):                        #和牌了
```

```
        Win_btn["state"]=NORMAL
        Out_btn["state"]=DISABLED     #“出牌”按钮无效
        showinfo(title="恭喜",message="玩家 Win!")
        return                        #玩家不需要再出牌
    Out_btn["state"]=NORMAL           #“出牌”按钮有效
    Get_btn["state"]=DISABLED         #“摸牌”按钮无效
    Chi_btn["state"]=DISABLED         #“吃牌”按钮无效
    Peng_btn["state"]=DISABLED        #“碰牌”按钮无效
    MyTurn=True
```

fun2()实现游戏过程中出牌顺序的控制逻辑。在游戏中有两个牌手，一个是玩家自己（0 号牌手），一个是计算机（1 号牌手）。在玩家出牌后系统自动调用 ComputerOut()实现计算机智能出牌，这时又轮到玩家出牌，需要判断计算机出的牌玩家是否可以吃、碰，如果可以，则“吃牌”“碰牌”按钮有效。

```
def  fun2():                          #出牌顺序控制
    MyTurn=True                       #轮到玩家出牌
    Get_btn["state"]=NORMAL           #“摸牌”按钮有效
    if(len(playersOutCard[1])>0):
        #取计算机出的牌，即最后一张
        card=playersOutCard[1][len(playersOutCard[1])-1]
        #判断计算机出的牌玩家是否可以吃、碰
        if(canPeng(playersCard[0],card)):    #玩家是否可以碰牌
            Peng_btn["state"]=NORMAL         #“碰牌”按钮有效
        if (canChi(playersCard[0],card)):    #玩家是否可以吃牌
            Chi_btn["state"]=NORMAL          #“吃牌”按钮有效
        #不能吃、碰则只能直接摸牌
        if ( not canChi(playersCard[0],card) and not canPeng(playersCard[0],card)):
            Peng_btn["state"]=DISABLED
            Chi_btn["state"]=DISABLED
            #OnBtnGet_Click();                #直接摸牌
    else:                                    #计算机没出过牌直接摸牌
        Get_btn["state"]=NORMAL              #“摸牌”按钮有效
```

为了实现在不能吃、碰的情况下自动摸牌，不需要等玩家单击“摸牌”按钮后才摸牌，可以将上面的“直接摸牌”行的注释取消掉，这样就可以减少让玩家摸牌的麻烦，但是如果可以选择吃、碰，这时还是可以让玩家单击“摸牌”按钮的，因为玩家可以放弃吃、碰。

ComputerOut(Order:int)实现计算机智能出牌，首先将 m_aCards[k]牌移动到对家（计算机）牌所在的位置，并按花色排序理牌。调用 ComputerCardNum(playersCard[0])计算牌手手中各种牌型的数量并判断是否和牌，如果和牌则游戏结束，否则调用 ComputerCard (playersCard[1])智能出牌。

```
def ComputerOut():         #计算机智能出牌
    global k,MyTurn
    #对家（计算机）摸牌
```

```
        m_aCards[k].MoveTo(90+55*13,80);
        m_aCards[k].setFront(True);           #显示麻将牌正面
        playersCard[1].append(m_aCards[k]); #第 14 张牌
        result1=ComputerCardNum(playersCard[1]);#计算计算机的各种牌型的数量，判断是否和牌
        if(result1):                          #和牌了
            showinfo(title="遗憾",message="计算机 Win!")
            return;                           #对家（计算机）不需要再出牌

        i=ComputerCard(playersCard[1]);       #智能出牌
        #i=0;                                 #总是出第 1 张牌，没有智能出牌
        card=playersCard[1][i]
        del(playersCard[1][i])
        #加到计算机出过牌的数组
        playersOutCard[1].append(card)
        #outCardOrder(playersOutCard[1]);     #整理出过的牌，Z 轴深度问题
        card.setFront(True);                  #显示麻将牌正面
        playSound(card)                       #根据计算机出牌选择声音文件播放
        #计算机按花色理手中的牌
        sortPoker2(playersCard[1]);
        card.x=len(playersOutCard[1])*25-25;
        card.y=10;
        card.MoveTo(card.x, card.y);
        k=k+1                                 #发过牌的总数
        MyTurn=True                           #轮到玩家
```

playSound(card)实现播放牌对应的声音文件。

```
def playSound(card):
    #music="res/sound/二条.wav";
    #根据牌的类型及编号来设置牌文件的路径及文件名
    music="res/sound/"+toChineseNumString(card.m nNum);
    if card.m_nType==1:          #桶（饼）
        music+="柄.wav";
    elif card.m_nType==2:        #条
        music+="条.wav";
    elif card.m_nType==3:        #万
        music+="万.wav";
    elif card.m_nType==4:        #字牌
        music="res/sound/give.wav";
    winsound.PlaySound(music, winsound.SND_FILENAME)
```

由于声音文件名是汉字，例如“一万.mp3”“二万.mp3”，所以在计算机出牌时toChineseNumString(n:int)将牌面的数字转换成汉字。

```
def toChineseNumString(n):
```

```
        if n==1:
            music="一"
        elif n==2:
            music="二"
        elif n==3:
            music="三"
        elif n==4:
            music="四"
        elif n==5:
            music="五"
        elif n==6:
            music="六"
        elif n==7:
            music="七"
        elif n==8:
            music="八"
        elif n==9:
            music="九"
        return music
```

在和牌算法中需要计算每种花色麻将牌的数量以及每种牌型的数量，ComputerCardNum(cards)根据 cards 计算出数据按和牌的数据结构存入 paiArray 中，调用和牌算法类中的 Win(paiArray)判断是否和牌。

```
    def ComputerCardNum(cards):     #玩家手中的牌
        #计算牌手手中各种牌型的数量
        paiArray=[[0,0,0,0,0,0,0,0,0,0],
                  [0,0,0,0,0,0,0,0,0,0],
                  [0,0,0,0,0,0,0,0,0,0],
                  [0,0,0,0,0,0,0,0,0,0]]
        print("玩家手中的牌",len(cards))
        for i in range(0,14):
            card=cards[i]
            if(card.imageID>10 and card.imageID<20):    #饼
                    paiArray[0][0]+=1
                    paiArray[0][card.imageID-10]+=1
            if(card.imageID>20 and card.imageID<30):    #条
                    paiArray[1][0]+=1
                    paiArray[1][card.imageID-20]+=1
            if(card.imageID>30 and card.imageID<40):    #万
                    paiArray[2][0]+=1
                    paiArray[2][card.imageID-30]+=1
            if(card.imageID>40 and card.imageID<50):    #字
                    paiArray[3][0]+=1
                    paiArray[3][card.imageID-40]+=1
        print(paiArray)
```

```
    hu=huMain()                    #和牌算法类
    result=hu.Win(paiArray)        #判断是否和牌
    return result
```

本两人麻将游戏还有许多地方需要完善，例如碰、吃牌功能，需要记录哪几张牌“吃”和“碰”，这几张牌不能再出，可以通过在 Card 类里增加 Selected 属性来记录是否用于“吃”和“碰”，这样玩家选择出牌时判断 Selected 属性的真假就可以知道是否能出。另外还有“杠”的处理，本游戏没有考虑，读者可以进一步去完善。本游戏的运行界面如图 12-3 所示。

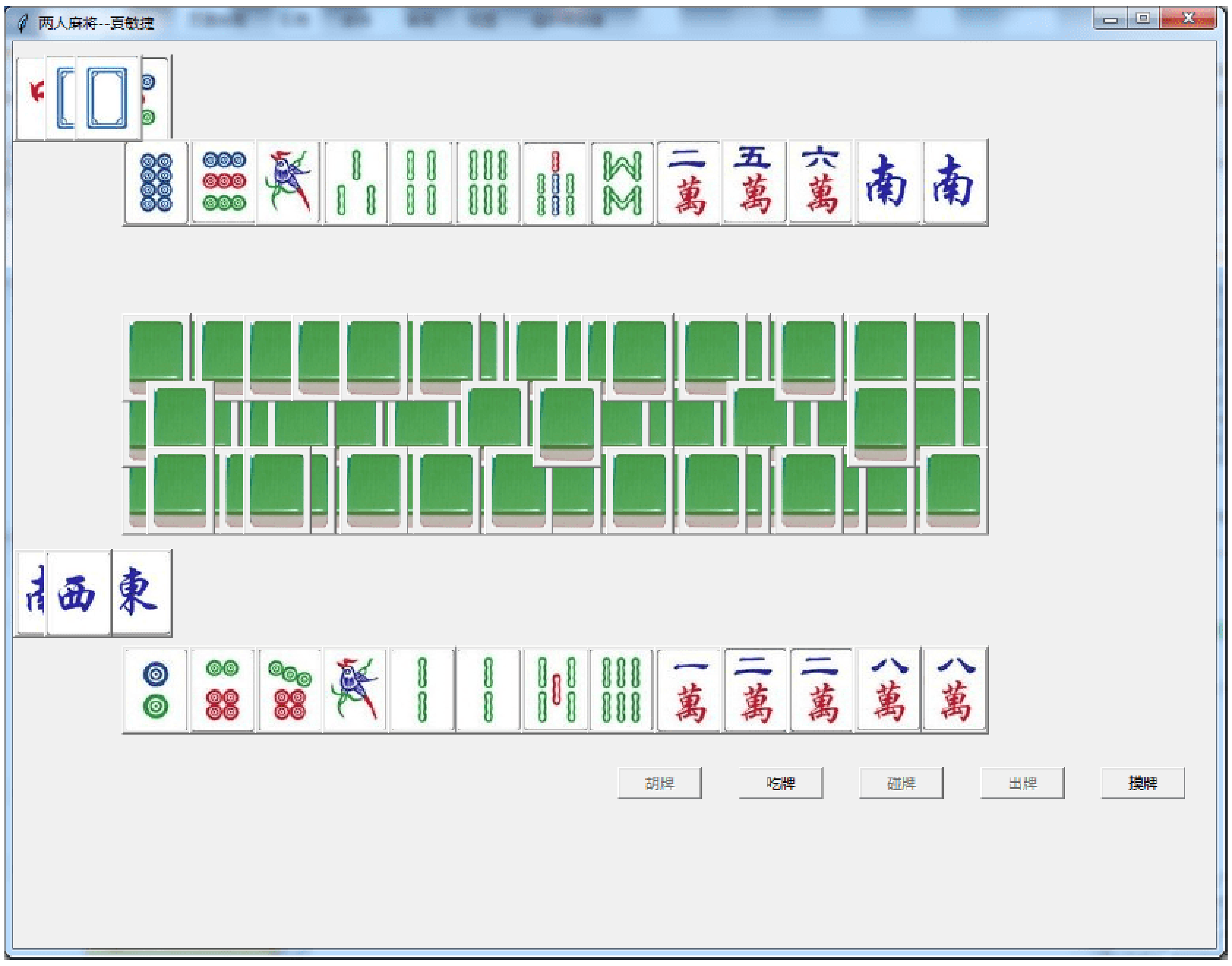


图 12-3　两人麻将游戏的运行界面

第 13 章

网络编程案例——基于TCP的在线聊天程序

视频讲解

13.1　基于 TCP 的在线聊天程序简介

本章基于 TCP 完成一个在线聊天程序，主要功能是实现客户端与服务器端的双向通信，运行效果如图 13-1 所示。

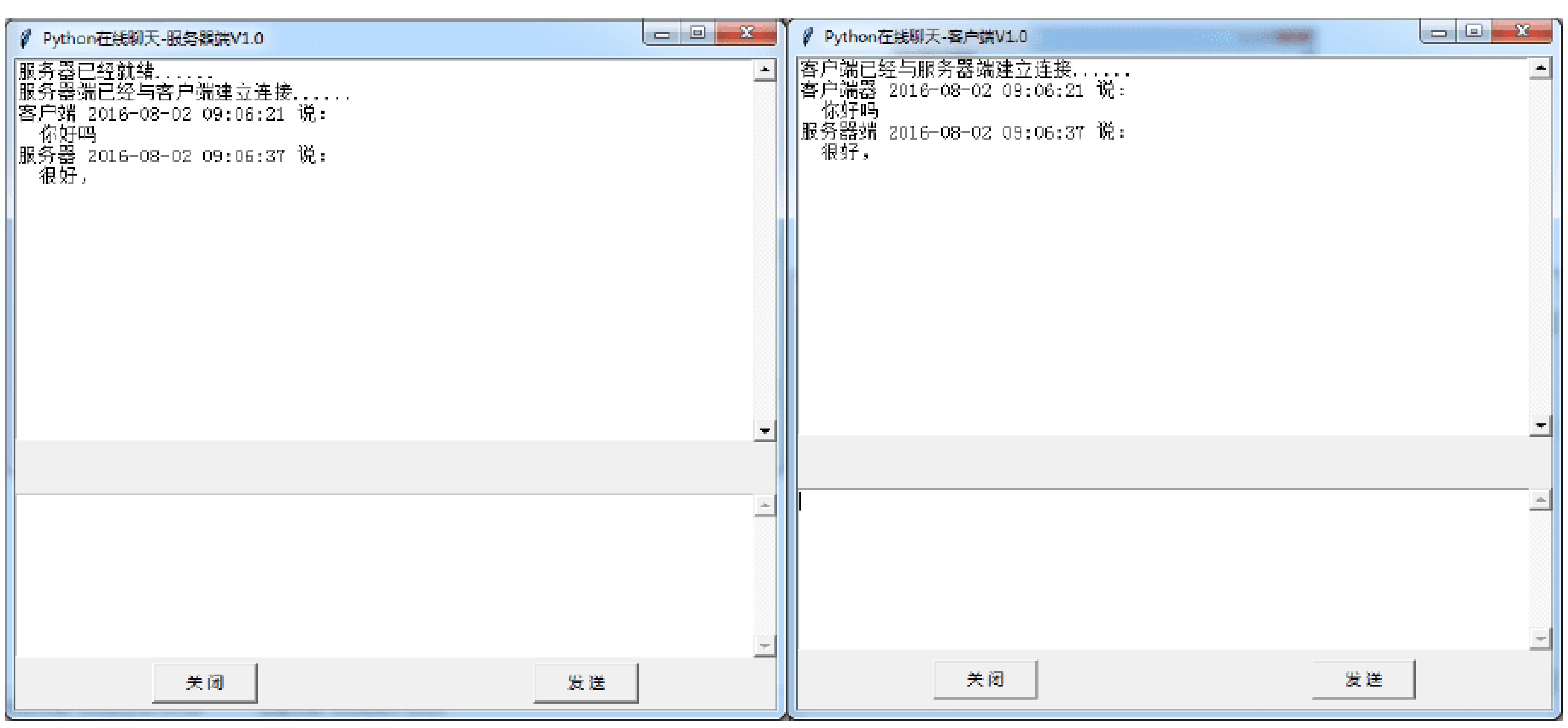


图 13-1　在线聊天的服务器端与客户端

13.2　关键技术

13.2.1　互联网 TCP/IP 协议

计算机为了连网，必须规定通信协议，早期的计算机网络都是由各厂商自己规定一套

协议，IBM、Apple 和 Microsoft 公司都有各自的网络协议，互不兼容，这就好比一群人有的说英语，有的说中文，有的说德语，说同一种语言的人可以交流，说不同语言的就不行了。

为了把全世界的所有不同类型的计算机都连接起来，必须规定一套全球通用的协议，为了实现互联网这个目标，国际组织制定了 OSI 七层模型互联网协议标准，如图 13-2 所示。因为互联网协议包含了上百种协议标准，但是其中最重要的两个协议是 TCP 和 IP 协议，所以大家把互联网协议简称 TCP/IP 协议。

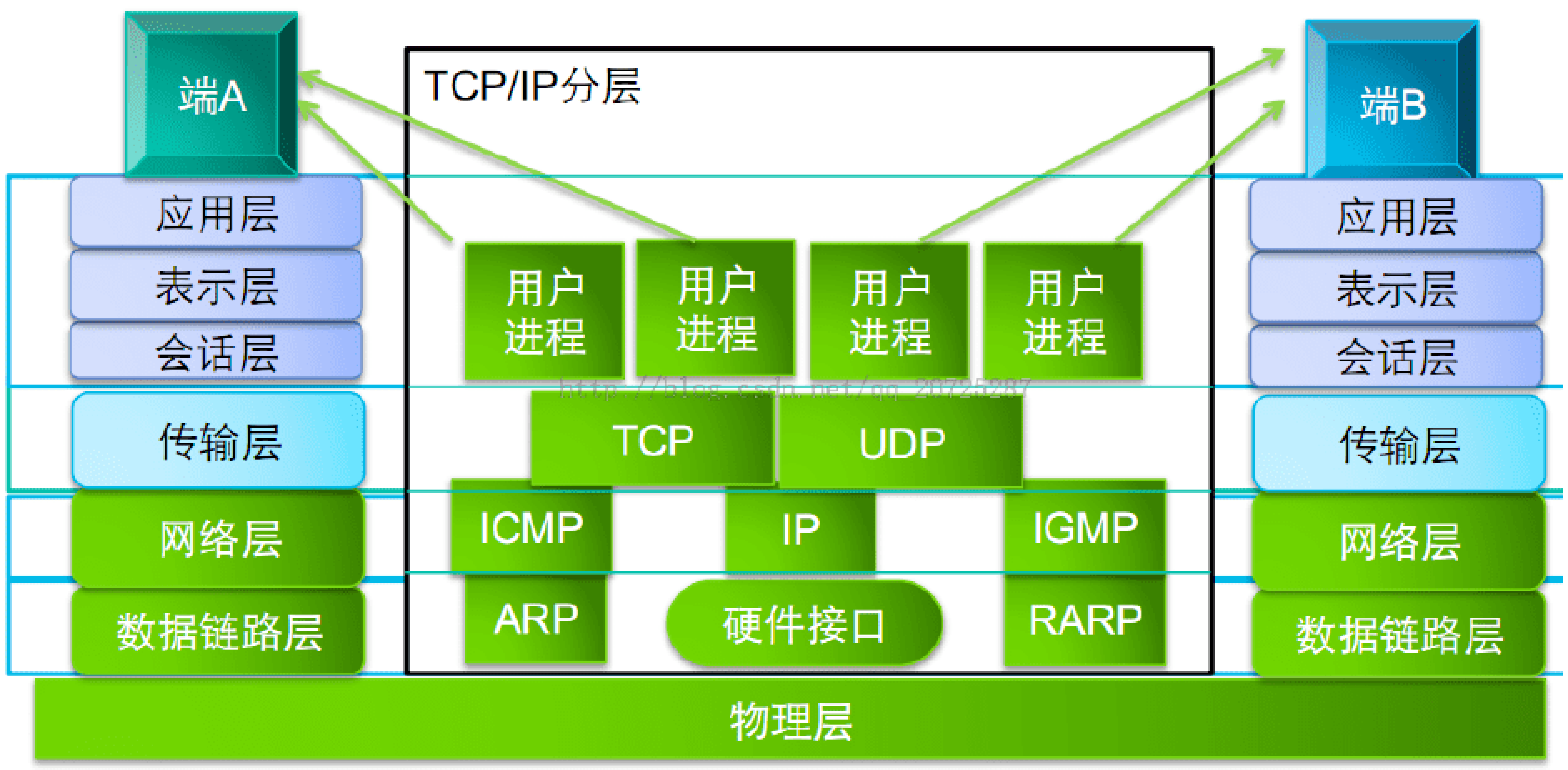


图 13-2　互联网协议

13.2.2　IP 协议和端口

❶ IP 协议

在通信的时候，双方必须知道对方的标识，这好比发邮件必须知道对方的邮件地址一样。互联网上每台计算机的唯一标识就是 IP 地址，类似于 202.196. 32.7。如果一台计算机同时接入到两个或更多的网络，例如路由器，它就会有两个或多个 IP 地址，所以 IP 地址对应的实际上是计算机的网络接口，通常是网卡。

IP 协议负责把数据从一台计算机通过网络发送到另一台计算机。数据被分割成一小块、一小块，然后通过 IP 包发送出去。由于互联网链路复杂，两台计算机之间经常有多条线路，因此路由器就负责决定如何把一个 IP 包转发出去。IP 包的特点是按块发送，途经多个路由，但不保证能到达，也不保证顺序到达。

IP 地址实际上是一个 32 位整数（称为 IPv4），以字符串表示的 IP 地址如 192.168.0.1，实际上是把 32 位整数按 8 位分组后的数字表示，目的是便于用户阅读。

IPv6 地址实际上是一个 128 位整数，它是目前使用的 IPv4 的升级版，以字符串表示，类似于 2001:0db8:85a3:0042:1000:8a2e:0370:7334。

❷ 端口

一个 IP 包除了包含要传输的数据外，还包含源 IP 地址和目标 IP 地址、源端口和目标

端口。

端口有什么作用？在两台计算机通信时只发 IP 地址是不够的，因为同一台计算机上运行着多个网络程序（例如浏览器、QQ 等网络程序）。在一个 IP 包来了之后，到底是交给浏览器还是 QQ，需要端口号来区分。每个网络程序都向操作系统申请唯一的端口号，这样两个进程在两台计算机之间建立网络连接就需要各自的 IP 地址和各自的端口号。例如浏览器经常使用 80 端口，FTP 程序使用 21 端口，邮件的收/发使用 25 端口。

网络上两台计算机之间的数据通信，归根结底就是不同主机的进程交互，而每个主机的进程又对应着某个端口。也就是说，单独靠 IP 地址是无法完成通信的，必须要有 IP 和端口。

13.2.3　TCP 协议和 UDP 协议

TCP 协议建立在 IP 协议之上。TCP 协议负责在两台计算机之间建立可靠连接，保证数据包按顺序到达。TCP 协议会通过握手建立连接，然后对每个 IP 包编号，确保对方按顺序收到，如果包丢掉了，就自动重发。

许多常用的更高级的协议都是建立在 TCP 协议基础上的，例如用于浏览器的 HTTP 协议、发送邮件的 SMTP 协议等。

UDP 协议同样建立在 IP 协议之上，但是 UDP 协议面向无连接的通信协议，不保证数据包的顺利到达，是不可靠传输，所以效率比 TCP 要高。

13.2.4　Socket

Socket 是网络编程的一个抽象概念。Socket 是套接字的英文名称，主要是用于网络通信编程。在 20 世纪 80 年代初，美国政府的高级研究工程机构（ARPA）给加利福尼亚大学的 Berkeley 分校提供了资金，让他们在 UNIX 操作系统下实现 TCP/IP 协议。在这个项目中，研究人员为 TCP/IP 网络通信开发了一个 API（应用程序接口），这个 API 称为 Socket（套接字）。Socket 是 TCP/IP 网络最为通用的 API，任何网络通信都是通过 Socket 来完成的。

通常用一个 Socket 表示“打开了一个网络链接”，而打开一个 Socket 需要知道目标计算机的 IP 地址和端口号，再指定协议类型。

套接字构造函数 socket(family,type[,protocol])使用给定的套接字家族、套接字类型、协议编号来创建套接字。

其参数如下。

- family：套接字家族，可以是 AF_UNIX 或者 AF_INET、AF_INET6。
- type：套接字类型，可以根据是面向连接的还是非连接的分为 SOCK_STREAM 和 SOCK_DGRAM。
- protocol：一般不填，默认为 0。

参数的取值的含义见表 13-1。

表 13-1　参数的取值含义

参　数	描　述
socket.AF_UNIX	只能够用于单一的 Unix 系统进程间通信
socket.AF_INET	服务器之间的网络通信
socket.AF_INET6	IPv6
socket.SOCK_STREAM	流式 Socket，针对 TCP
socket.SOCK_DGRAM	数据报式 Socket，针对 UDP
socket.SOCK_RAW	原始套接字，首先，普通的套接字无法处理 ICMP、IGMP 等网络报文，而 SOCK_RAW 可以；其次，SOCK_RAW 也可以处理特殊的 IPv4 报文；此外，利用原始套接字，可以通过 IP_HDRINCL 套接字选项由用户构造 IP 头
socket.SOCK_SEQPACKET	可靠的连续数据包服务

例如创建 TCP Socket：

```
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
```

创建 UDP Socket：

```
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
```

Socket 同时支持数据流 Socket 和数据报 Socket。图 13-3 是面向连接支持数据流 TCP 的时序图。

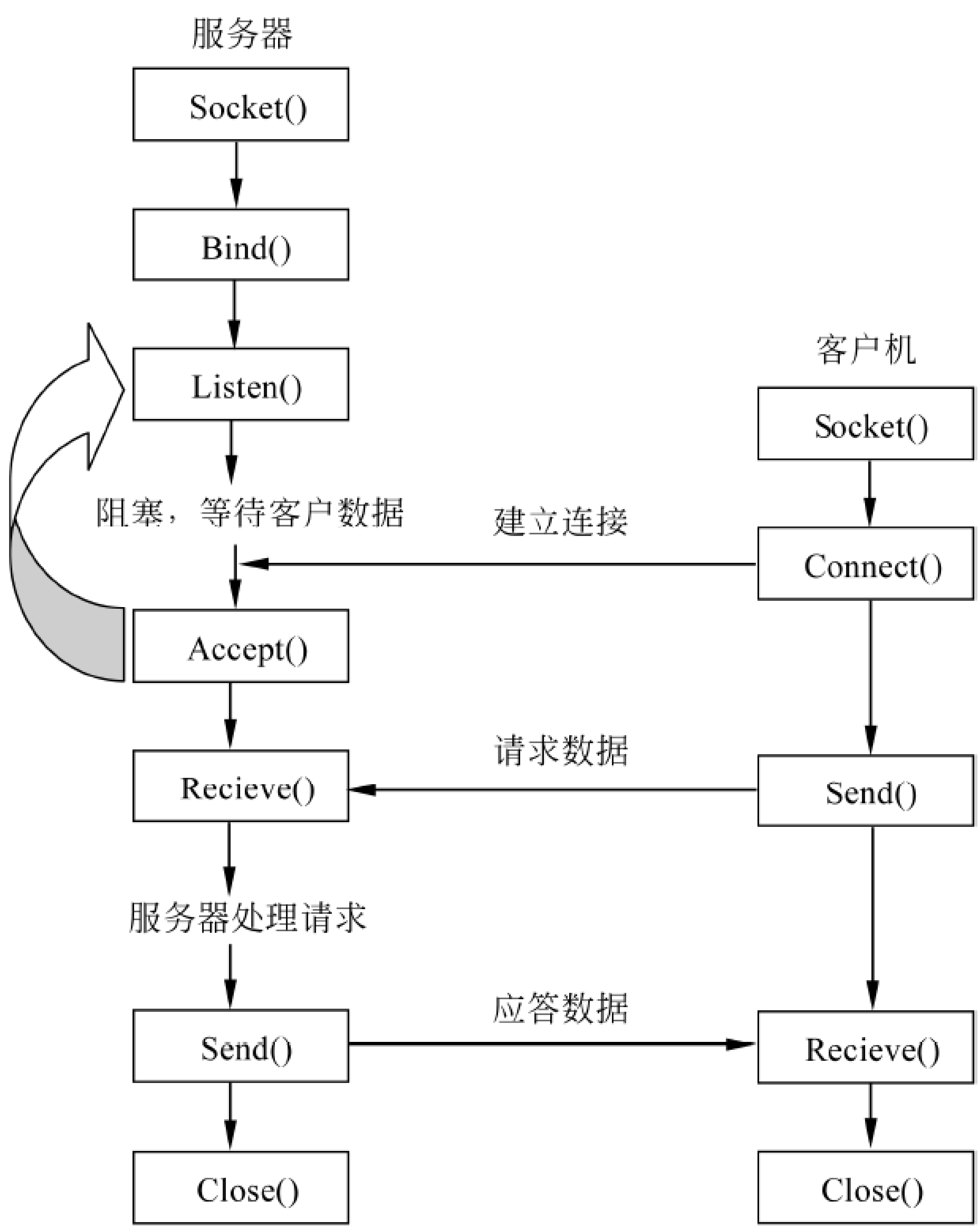


图 13-3　面向连接 TCP 的时序图

由该图可以看出，客户机（Client）与服务器（Server）的关系是不对称的。

对于 TCP C/S，服务器首先启动，然后在某一时刻启动客户机与服务器建立连接。服务器与客户机开始都必须调用 Socket()建立一个套接字，然后服务器调用 Bind()将套接字与一个本机指定端口绑定在一起，再调用 Listen()使套接字处于一种被动的准备接收状态，这时客户机建立套接字便可以通过调用 Connect()和服务器建立连接，服务器就可以调用 Accept()来接收客户机连接。然后继续监听指定端口，并发出阻塞，直到下一个请求出现，从而实现多个客户机连接。在连接建立之后，客户机和服务器之间就可以通过连接发送和接收数据。最后，待数据传送结束，双方调用 Close()关闭套接字。

在 Python 的 Socket 模块中 Socket 对象提供的函数如表 13-2 所示。

表 13-2　Socket 对象的函数

函　　数	描　　述
服务器端套接字函数	
s.bind(host,port)	绑定地址（host,port）到套接字，在 AF_INET 下以元组（host,port）的形式表示地址
s.listen(backlog)	开始 TCP 监听。backlog 指定在拒绝连接之前可以的最大连接数量。该值至少为 1，大部分应用程序设为 5 就可以了
s.accept()	被动接受 TCP 客户端连接，（阻塞式）等待连接的到来
客户端套接字函数	
s.connect(address)	主动与 TCP 服务器连接。一般 address 的格式为元组（hostname,port），如果连接出错，返回 socket.error 错误
s.connect_ex()	connect()函数的扩展版本，出错时返回出错码，而不是抛出异常
公共用途的套接字函数	
s.recv(bufsize,[,flag])	接收 TCP 数据，数据以字节串形式返回。bufsize 指定要接收的最大数据量。flag 提供有关消息的其他信息，通常可以忽略
s.send(data)	发送 TCP 数据，将 data 中的数据发送到连接的套接字。其返回值是要发送的字节数量，该数量可能小于 data 的字节大小
s.sendall(data)	完整发送 TCP 数据，将 data 中的数据发送到连接的套接字，但在返回之前会尝试发送所有数据。如果成功，返回 None，如果失败，抛出异常
s.recvform(bufsize,[,flag])	接收 UDP 数据，与 recv()类似，但返回值是（data,address）。其中，data 是包含接收数据的字节串，address 是发送数据的套接字地址
s.sendto(data,address)	发送 UDP 数据，将数据发送到套接字，address 是形式为（ip,port）的元组，指定远程地址。其返回值是发送的字节数
s.close()	关闭套接字
s.getpeername()	返回连接套接字的远程地址，其返回值通常是元组（ipaddr,port）
s.getsockname()	返回套接字自己的地址，通常是一个元组（ipaddr,port）
s.setsockopt(level, optname,value)	设置给定套接字选项的值
s.getsockopt(level, optname)	返回套接字选项的值
s.settimeout(timeout)	设置套接字操作的超时时间，timeout 是一个浮点数，单位是秒。其值为 None 表示没有超时时间。一般情况下，超时时间应该在刚创建套接字时设置，因为它们可能用于连接操作（例如 connect()）
s.gettimeout()	返回当前超时时间的值，单位是秒，如果没有设置超时期，则返回 None
s.fileno()	返回套接字的文件描述符

续表

函　数	描　述
s.setblocking(flag)	如果 flag 为 0，则将套接字设为非阻塞模式，否则将套接字设为阻塞模式（默认值）。在非阻塞模式下，如果调用 recv()没有发现任何数据，或 send()调用无法立即发送数据，那么将引起 socket.error 异常
s.makefile()	创建一个与该套接字相关联的文件

在了解了 TCP/IP 协议的基本概念以及 IP 地址、端口的概念和 Socket 之后，就可以开始进行网络编程了。

例 13-1 编写一个简单的 TCP 服务器程序，它接收客户端连接，把客户端发过来的字符串加上"Hello"再发回去。

完整的 TCP 服务器端程序如下：

```
import socket                                          #导入 socket 模块
import threading                                       #导入 threading 线程模块
def tcplink(sock, addr):
    print('接收一个来自%s:%s 连接请求' % addr)
    sock.send(b'Welcome!')                             #发给客户端 Welcome!信息
    while True:
        data=sock.recv(1024)                           #接收客户端发来的信息
        time.sleep(1)                                  #延时 1 秒钟
        if not data or data.decode('utf-8')=='exit':   #如果没数据或收到“exit”信息
            break                                      #终止循环
        sock.send(('Hello, %s!' % data.decode('utf-8')).encode('utf-8'))
                                                       #收到信息加上 Hello 发回
    sock.close()                                       #关闭连接
    print('来自 %s:%s 连接关闭了.' % addr)
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('127.0.0.1', 8888))                            #监听本机 8888 端口
s.listen(5)                                            #连接的最大数量为 5
print('等待客户端连接...')
while True:
    sock, addr=s.accept()                              #接受一个新连接
    #创建新线程来处理 TCP 连接
    t=threading.Thread(target=tcplink, args=(sock, addr))
    t.start()
```

在程序中首先创建一个基于 IPv4 和 TCP 协议的 Socket：

```
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

然后绑定监听的地址和端口。服务器可能有多块网卡，可以绑定到某一块网卡的 IP 地址上，也可以用 0.0.0.0 绑定到所有的网络地址，还可以用 127.0.0.1 绑定到本机地址。127.0.0.1 是一个特殊的 IP 地址，表示本机地址，如果绑定到这个地址，客户端必须同时在本机运行才能连接，也就是说，外部的计算机无法连接进来。

端口号需要预先指定。因为这里写的这个服务不是标准服务，所以用 8888 这个端口

号。注意，小于 1024 的端口号必须要有管理员权限才能绑定。

```
#监听本机 8888 端口
s.bind(('127.0.0.1', 8888))
```

接着调用 listen()方法开始监听端口，传入的参数指定等待连接的最大数量为 5：

```
s.listen(5)
print('等待客户端连接...')
```

接下来，服务器程序通过一个无限循环接受来自客户端的连接，accept()会等待并返回一个客户端的连接。

```
while True:
    #接受一个新连接
    sock, addr=s.accept()#sock 是新建的 Socket 对象，服务器通过它与对应客户端通
                         #信，addr 是 IP 地址
    #创建新线程来处理 TCP 连接
    t=threading.Thread(target=tcplink, args=(sock, addr))
    t.start()
```

每个连接都必须创建新线程（或进程）来处理，否则单线程在处理连接的过程中无法接受其他客户端的连接：

```
def tcplink(sock, addr):
    print('接收一个来自%s:%s 连接请求' % addr)
    sock.send(b'Welcome!')                          #发给客户端“Welcome!”信息
    while True:
        data=sock.recv(1024)                        #接收客户端发来的信息
        time.sleep(1)                               #延时 1 秒钟
        if not data or data.decode('utf-8')=='exit':#如果没数据或收到'exit'信息
            break                                   #终止循环
        sock.send(('Hello, %s!' % data.decode('utf-8')).encode('utf-8'))
                                                    #收到信息加上“Hello”发回
    sock.close()                                    #关闭连接
    print('来自 %s:%s 连接关闭了.' % addr)
```

在连接建立后，服务器首先发一条欢迎消息，然后等待客户端数据，并加上“Hello”再发送给客户端。如果客户端发送了 exit 字符串，就直接关闭连接。

如果要测试这个服务器程序，还需要编写一个客户端程序：

```
import socket                                       #导入 socket 模块
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('127.0.0.1', 8888))                      #建立连接
#打印接收到欢迎消息
print(s.recv(1024).decode('utf-8'))
for data in [b'Michael', b'Tracy', b'Sarah']:
    s.send(data)                                #客户端程序发送人名数据给服务器端
    print(s.recv(1024).decode('utf-8'))
```

```
    s.send(b'exit')
    s.close()
```

打开两个命令行窗口，一个是运行服务器端程序，另一个是运行客户端程序，就可以看到运行效果，如图 13-4 和图 13-5 所示。

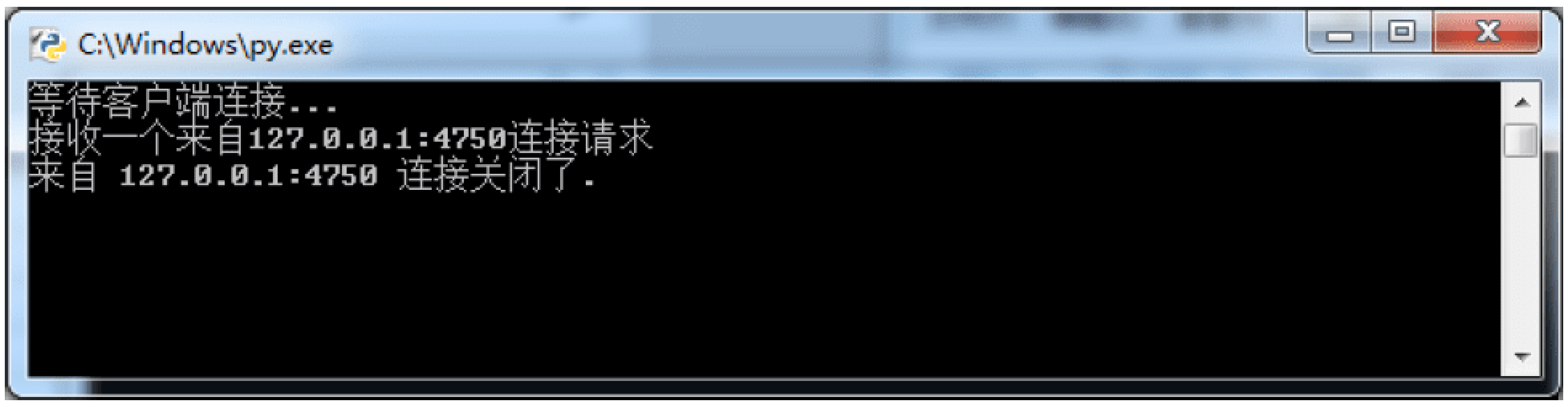


图 13-4　服务器程序效果

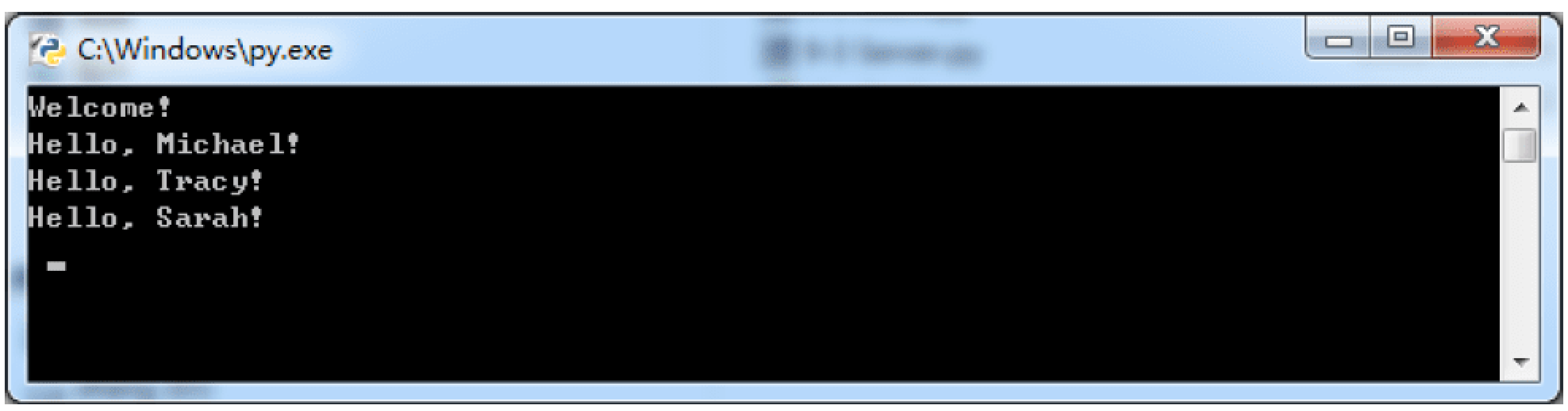


图 13-5　客户端程序效果

需要注意的是，客户端程序运行完毕就退出了，而服务器程序会永远运行下去，必须按 Ctrl+C 组合键退出程序。

可见，用 TCP 协议进行 Socket 编程在 Python 中十分简单，对于客户端，要主动连接服务器的 IP 和指定端口，对于服务器，要首先监听指定端口，然后对每一个新的连接创建一个线程或进程来处理。通常，服务器程序会无限运行下去。另外还需注意，同一个端口被一个 Socket 绑定了以后就不能被其他 Socket 绑定了。

13.2.5　多线程编程

线程是操作系统可以调度的最小执行单位，能够执行并发处理。通常是将程序拆分成两个或多个并发运行的线程，即同时执行多个操作。例如，在使用线程的同时监视用户并发输入，并执行后台任务等。

threading 模块提供了 Thread 类来创建和处理线程，格式如下：

```
线程对象=threading.Thread(target=线程函数,args=(参数列表), name=线程名,
group=线程组)
```

第一个参数是函数名，第二个参数 args 是一个元组，线程名和线程组都可以省略。

Thread 类还提供了以下方法。

- run()：用于表示线程活动的方法。
- start()：启动线程活动。
- join([time])：可以阻塞进程，直到线程执行完毕。参数 time 指定超时时间（单位为秒），超过指定时间 join 就不再阻塞进程了。
- isAlive()：返回线程是否活动。
- getName()：返回线程名。
- setName()：设置线程名。

threading 模块提供的其他方法如下：

- threading.currentThread()：返回当前的线程变量。
- threading.enumerate()：返回一个包含正在运行的线程的 list。正在运行指线程启动后、结束前，不包括启动前和终止后的线程。
- threading.activeCount()：返回正在运行的线程数量，与 len(threading.enumerate())有相同的结果。

例 13-2　编写自己的线程类 myThread 来创建线程对象。

分析：自己的线程类直接从 threading.Thread 类继承，然后重写__init__()方法和 run()方法就可以创建线程对象了。

```
import threading
import time
exitFlag=0
class myThread(threading.Thread):   #继承父类 threading.Thread
    def __init__(self, threadID, name, counter):
        threading.Thread.__init__(self)
        self.threadID=threadID
        self.name=name
        self.counter=counter
    def run(self):#把要执行的代码写到 run()函数里面，线程在创建后会直接运行 run()函数
        print("Starting "+self.name)
        print time(self.name, self.counter, 5)
        print("Exiting "+self.name)
def print time(threadName, delay, counter):
    while counter:
        if exitFlag:
            thread.exit()
        time.sleep(delay)
        print("%s: %s"%(threadName, time.ctime(time.time())))
        counter-=1
#创建新线程
thread1=myThread(1, "Thread-1", 1)
thread2=myThread(2, "Thread-2", 2)
```

```
#开启线程
thread1.start()
thread2.start()
print("Exiting Main Thread")
```

以上程序的执行结果如下：

```
Starting Thread-1  Exiting Main Thread  Starting Thread-2
Thread-1: Tue Aug  2 10:19:01 2016
Thread-2: Tue Aug  2 10:19:02 2016
Thread-1: Tue Aug  2 10:19:02 2016
Thread-1: Tue Aug  2 10:19:03 2016
Thread-2: Tue Aug  2 10:19:04 2016
Thread-1: Tue Aug  2 10:19:04 2016
Thread-1: Tue Aug  2 10:19:05 2016
Exiting Thread-1
Thread-2: Tue Aug  2 10:19:06 2016
Thread-2: Tue Aug  2 10:19:08 2016
Thread-2: Tue Aug  2 10:19:10 2016
Exiting Thread-2
```

13.3　在线聊天程序设计的步骤

视频讲解

13.3.1　在线聊天程序的服务器端

在服务器端设计 ServerUI 类，封装接收消息函数方法 receiveMessage(self)、发送消息 sendMessage(self)，并在构造函数中完成 Tkinter 界面布局。

在服务器端建立 Socket 并绑定 5505 后循环接受客户端的连接请求。当服务器与客户端的连接建立后，如果客户端发送字符 Y，服务器端收到后会返回字符 Y 信息，表明连接建立成功。在连接建立成功后即可不断接收客户端发来的聊天信息。

下面是服务器端代码：

```
#Filename:ServerUI.py
#Python 在线聊天服务器端
import tkinter
import tkinter.font as tkFont
import socket
import threading
import time,tsys
class ServerUI():
    local='127.0.0.1'
    port=5505
    global serverSock;
```

```
    flag=False
    #初始化类的相关属性的构造函数
    def __init__(self):
        self.root=tkinter.Tk()
        self.root.title('Python 在线聊天-服务器端 V1.0')
        #窗口面板，用 4 个 frame 面板布局
        self.frame=[tkinter.Frame(),tkinter.Frame(),tkinter.Frame(),
        tkinter.Frame()]
        #显示消息 Text 右边的滚动条
        self.chatTextScrollBar=tkinter.Scrollbar(self.frame[0])
        self.chatTextScrollBar.pack(side=tkinter.RIGHT,fill=tkinter.Y)
        #显示消息 Text，并绑定上面的滚动条
        ft=tkFont.Font(family='Fixdsys',size=11)
        self.chatText=tkinter.Listbox(self.frame[0],width=70,height=18,
        font=ft)
        self.chatText['yscrollcommand']=self.chatTextScrollBar.set
        self.chatText.pack(expand=1,fill=tkinter.BOTH)
        self.chatTextScrollBar['command']=self.chatText.yview()
        self.frame[0].pack(expand=1,fill=tkinter.BOTH)
        #标签，分开消息显示 Text 和消息输入 Text
        label=tkinter.Label(self.frame[1],height=2)
        label.pack(fill=tkinter.BOTH)
        self.frame[1].pack(expand=1,fill=tkinter.BOTH)
        #输入消息 Text 的滚动条
        self.inputTextScrollBar=tkinter.Scrollbar(self.frame[2])
        self.inputTextScrollBar.pack(side=tkinter.RIGHT,fill=tkinter.Y)
        #输入消息 Text，并与滚动条绑定
        ft=tkFont.Font(family='Fixdsys',size=11)
        self.inputText=tkinter.Text(self.frame[2],width=70,height=8,font=ft)
        self.inputText['yscrollcommand']=self.inputTextScrollBar.set
        self.inputText.pack(expand=1,fill=tkinter.BOTH)
        self.inputTextScrollBar['command']=self.chatText.yview()
        self.frame[2].pack(expand=1,fill=tkinter.BOTH)
        #“发送”按钮
        self.sendButton=tkinter.Button(self.frame[3],text='发送',width=10,
        command=self.sendMessage)
        self.sendButton.pack(expand=1,side=tkinter.BOTTOM and tkinter
        .RIGHT,padx=25,pady=5)
        #“关闭”按钮
        self.closeButton=tkinter.Button(self.frame[3],text='关闭',width=10,command
        =self.close)
        self.closeButton.pack(expand=1,side=tkinter.RIGHT,padx=25,pady=5)
        self.frame[3].pack(expand=1,fill=tkinter.BOTH)
```

```
    #接收消息
    def receiveMessage(self):
        #建立 Socket 连接
        self.serverSock=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
        self.serverSock.bind((self.local,self.port))
        self.serverSock.listen(15)
        self.buffer=1024
        self.chatText.insert(tkinter.END,'服务器已经就绪......')
        #循环接受客户端的连接请求
        while True:
            self.connection,self.address=self.serverSock.accept()
            self.flag=True
            while True:
                #接收客户端发送的消息
                self.cientMsg=self.connection.recv(self.buffer).decode('utf-8')
                if not self.cientMsg:
                    continue
                elif self.cientMsg=='Y':
                    self.chatText.insert(tkinter.END,'服务器端已经与客户端建立
                    连接......')
                    self.connection.send(b'Y')
                elif self.cientMsg=='N':
                    self.chatText.insert(tkinter.END,'服务器端与客户端建立连接
                    失败......')
                    self.connection.send(b'N')
                else:
                    theTime=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
                    self.chatText.insert(tkinter.END, '客户端 ' + theTime +' 说: \n')
                    self.chatText.insert(tkinter.END, '  ' + self.cientMsg)
    #发送消息
    def sendMessage(self):
        #得到用户在 Text 中输入的消息
        message=self.inputText.get('1.0',tkinter.END)
        #格式化当前的时间
        theTime=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
        self.chatText.insert(tkinter.END, '服务器 '+theTime+'说: \n')
        self.chatText.insert(tkinter.END,'  '+message+'\n')
        if self.flag==True:
            #将消息发送到客户端
            self.connection.send(message.encode())
        else:
            #Socket 连接没有建立，提示用户
            self.chatText.insert(tkinter.END,'您还未与客户端建立连接，客户端无法
```

```
            收到您的消息\n')
            #清空用户在 Text 中输入的消息
            self.inputText.delete(0.0,message.__len__()-1.0)
        #关闭消息窗口并退出
        def close(self):
            sys.exit()
        #启动线程接收客户端的消息
        def startNewThread(self):
            #启动一个新线程来接收客户端的消息
            #args 是传递给线程函数的参数，receiveMessage 函数不需要参数，只传一个空元组
            thread=threading.Thread(target=self.receiveMessage,args=())
            thread.setDaemon(True);
            thread.start();
    def main():
        server=ServerUI()
        server.startNewThread()
        server.root.mainloop()
    if __name__=='__main__':
        main()
```

13.3.2　在线聊天程序的客户端

在客户端设计 ClientUI 类，封装接收消息函数方法 receiveMessage(self)、发送消息 sendMessage(self)，并在构造函数中完成 Tkinter 界面布局。

在客户端建立 Socket 后，向服务器发送字符 Y，表示客户端要连接服务器。服务器端收到后会返回字符 Y 信息，表明连接建立成功。在连接建立成功后即可不断接收服务器发来的聊天信息。

下面是客户端代码：

```
    #Filename:ClientUI.py
    #Python 在线聊天客户端  2016-2-12
    import tkinter
    import tkinter.font as tkFont
    import socket
    import threading
    import time,tsys
    class ClientUI():
        local='127.0.0.1'
        port=5505
        global clientSock;
        flag=False
```

```
    #初始化类的相关属性的构造函数
    def __init__(self):
        self.root=tkinter.Tk()
        self.root.title('Python 在线聊天-客户端 V1.0')
        #窗口面板，用 4 个面板布局
        self.frame=[tkinter.Frame(),tkinter.Frame(),tkinter.Frame(),
        tkinter.Frame()]
        #以下界面设计与服务器端相同
        #显示消息 Text 右边的滚动条
        self.chatTextScrollBar=tkinter.Scrollbar(self.frame[0])
        self.chatTextScrollBar.pack(side=tkinter.RIGHT,fill=tkinter.Y)
        #显示消息 Text，并绑定上面的滚动条
        ft=tkFont.Font(family='Fixdsys',size=11)
        self.chatText=tkinter.Listbox(self.frame[0],width=70,height=18,font=ft)
        self.chatText['yscrollcommand']=self.chatTextScrollBar.set
        self.chatText.pack(expand=1,fill=tkinter.BOTH)
        self.chatTextScrollBar['command']=self.chatText.yview()
        self.frame[0].pack(expand=1,fill=tkinter.BOTH)
        #标签，分开消息显示 Text 和消息输入 Text
        label=tkinter.Label(self.frame[1],height=2)
        label.pack(fill=tkinter.BOTH)
        self.frame[1].pack(expand=1,fill=tkinter.BOTH)
        #输入消息 Text 的滚动条
        self.inputTextScrollBar=tkinter.Scrollbar(self.frame[2])
        self.inputTextScrollBar.pack(side=tkinter.RIGHT,fill=tkinter.Y)
        #输入消息 Text，并与滚动条绑定
        ft=tkFont.Font(family='Fixdsys',size=11)
        self.inputText=tkinter.Text(self.frame[2],width=70,height=8,font=ft)
        self.inputText['yscrollcommand']=self.inputTextScrollBar.set
        self.inputText.pack(expand=1,fill=tkinter.BOTH)
        self.inputTextScrollBar['command']=self.chatText.yview()
        self.frame[2].pack(expand=1,fill=tkinter.BOTH)
        #“发送”按钮
        self.sendButton=tkinter.Button(self.frame[3],text='发送',
        width=10,command=self.sendMessage)
        self.sendButton.pack(expand=1,side=tkinter.BOTTOM and tkinter
        .RIGHT,padx=15,pady=8)
        #“关闭”按钮
        self.closeButton=tkinter.Button(self.frame[3],text='关闭',width=10,
        command=self.close)
        self.closeButton.pack(expand=1,side=tkinter.RIGHT,padx=15,pady=8)
        self.frame[3].pack(expand=1,fill=tkinter.BOTH)
```

```
    #接收消息
    def receiveMessage(self):
        try:
            #建立 Socket 连接
            self.clientSock=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
            self.clientSock.connect((self.local, self.port))
            self.flag=True
        except:
            self.flag=False
            self.chatText.insert(tkinter.END,'您还未与服务器端建立连接，请检查服
            务器是否启动')
            return
        self.buffer=1024
        self.clientSock.send('Y'.encode()) #向服务器发送字符 Y，表示客户端要连接服务器
        while True:
            try:
                if self.flag==True:
                    #连接建立，接收服务器端消息
                    self.serverMsg=self.clientSock.recv(self.buffer)
                    .decode('utf-8')
                    if self.serverMsg=='Y':
                        self.chatText.insert(tkinter.END,'客户端已经与服务器端
                        建立连接......')
                    elif self.serverMsg=='N':
                        self.chatText.insert(tkinter.END,'客户端与服务器端建立
                        连接失败......')
                    elif not self.serverMsg:
                        continue
                    else:
                        theTime=time.strftime("%Y-%m-%d %H:%M:%S", time
                        .localtime())
                        self.chatText.insert(tkinter.END, '服务器端 ' + theTime
                        +' 说: \n')
                        self.chatText.insert(tkinter.END, '  ' + self.serverMsg)
                else:
                    break
            except EOFError as msg:
                raise msg
                self.clientSock.close()
                break
    #发送消息
    def sendMessage(self):
```

```
            #得到用户在 Text 中输入的消息
            message=self.inputText.get('1.0',tkinter.END)
            #格式化当前的时间
            theTime=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime())
            self.chatText.insert(tkinter.END, '客户端器 ' + theTime +' 说: \n')
            self.chatText.insert(tkinter.END,'  ' + message + '\n')
            if self.flag==True:
                self.clientSock.send(message.encode());      #将消息发送到服务器端
            else:
                #Socket 连接没有建立，提示用户
                self.chatText.insert(tkinter.END,'您还未与服务器端建立连接，服务器端
                无法收到您的消息\n')
            #清空用户在 Text 中输入的消息
            self.inputText.delete(0.0,message.__len__()-1.0)
        #关闭消息窗口并退出
        def close(self):
            sys.exit()
        #启动线程接收服务器端的消息
        def startNewThread(self):
            #启动一个新线程来接收服务器端的消息
            #args 是传递给线程函数的参数，receiveMessage 函数不需要参数，只传一个空元组
            thread=threading.Thread(target=self.receiveMessage,args=())
            thread.setDaemon(True);
            thread.start();
    def main():
        client=ClientUI()
        client.startNewThread()          #启动线程接收服务器端的消息
        client.root.mainloop()
    if __name__=='__main__':
        main()
```

第 14 章

网络通信案例——基于 UDP 的网络五子棋游戏

14.1　网络五子棋游戏简介

五子棋是一种家喻户晓的棋类游戏，它的多变吸引了无数玩家。本章设计的五子棋游戏是一简易五子棋，棋盘为 15×15，黑子先落。在每次下棋子前先判断该处有无棋子，有则不能落子，超出边界不能落子。任何一方有达到横向、竖向、斜向、反斜向连到 5 个棋子则胜利。

本章介绍基于 UDP 的 Socket 编程方法来制作网络五子棋游戏程序。网络五子棋游戏采用 C/S 架构，分为服务器端和客户端。服务器端运行界面如图 14-1 所示，在游戏时服务器端首先启动，当客户端连接后服务器端可以走棋。

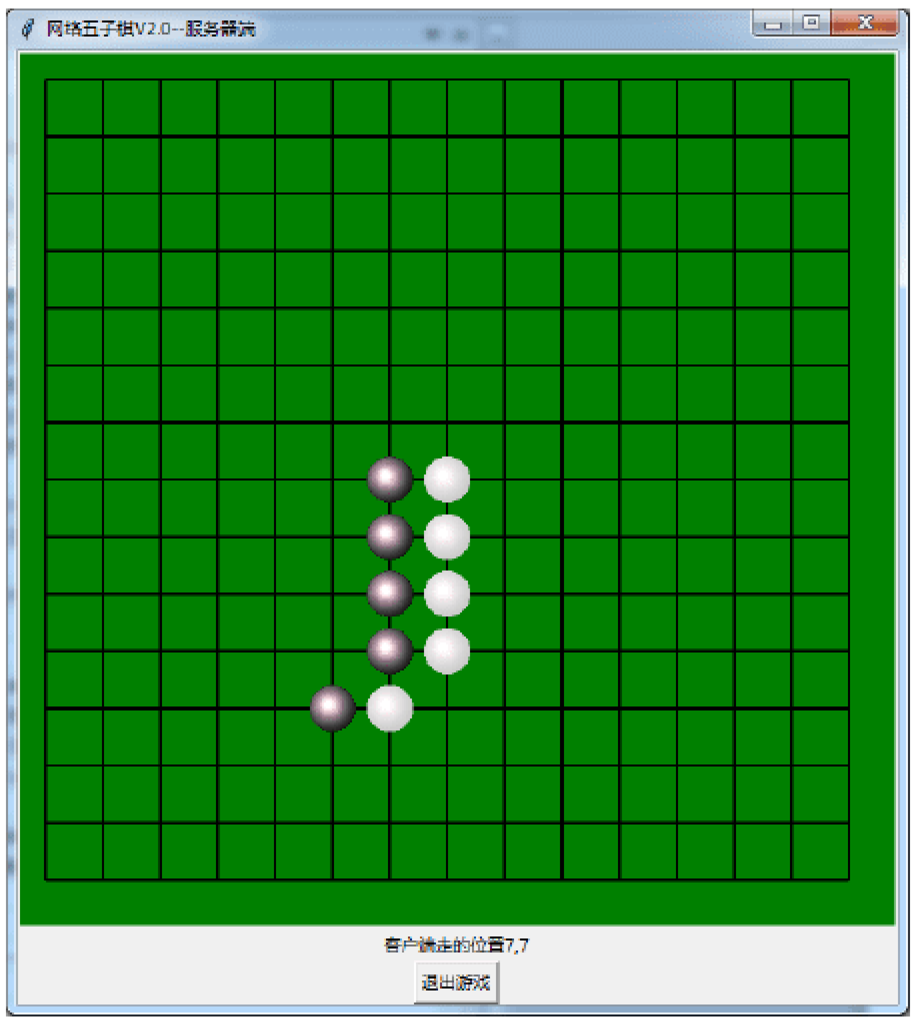


图 14-1　网络五子棋游戏的服务器端界面

用户根据提示信息，轮到自己下棋才可以在棋盘上落子，同时下方标签会显示对方的走棋信息，服务器端用户通过“退出游戏”按钮结束游戏。

客户端运行界面如图 14-2 所示，需要输入服务器的 IP 地址（这里采用默认地址，即本机地址），如果正确且服务器启动则可以连接服务器。连接成功后客户端用户根据提示信息，轮到自己下棋才可以在棋盘上落子，同样可以通过“退出游戏”按钮结束游戏。

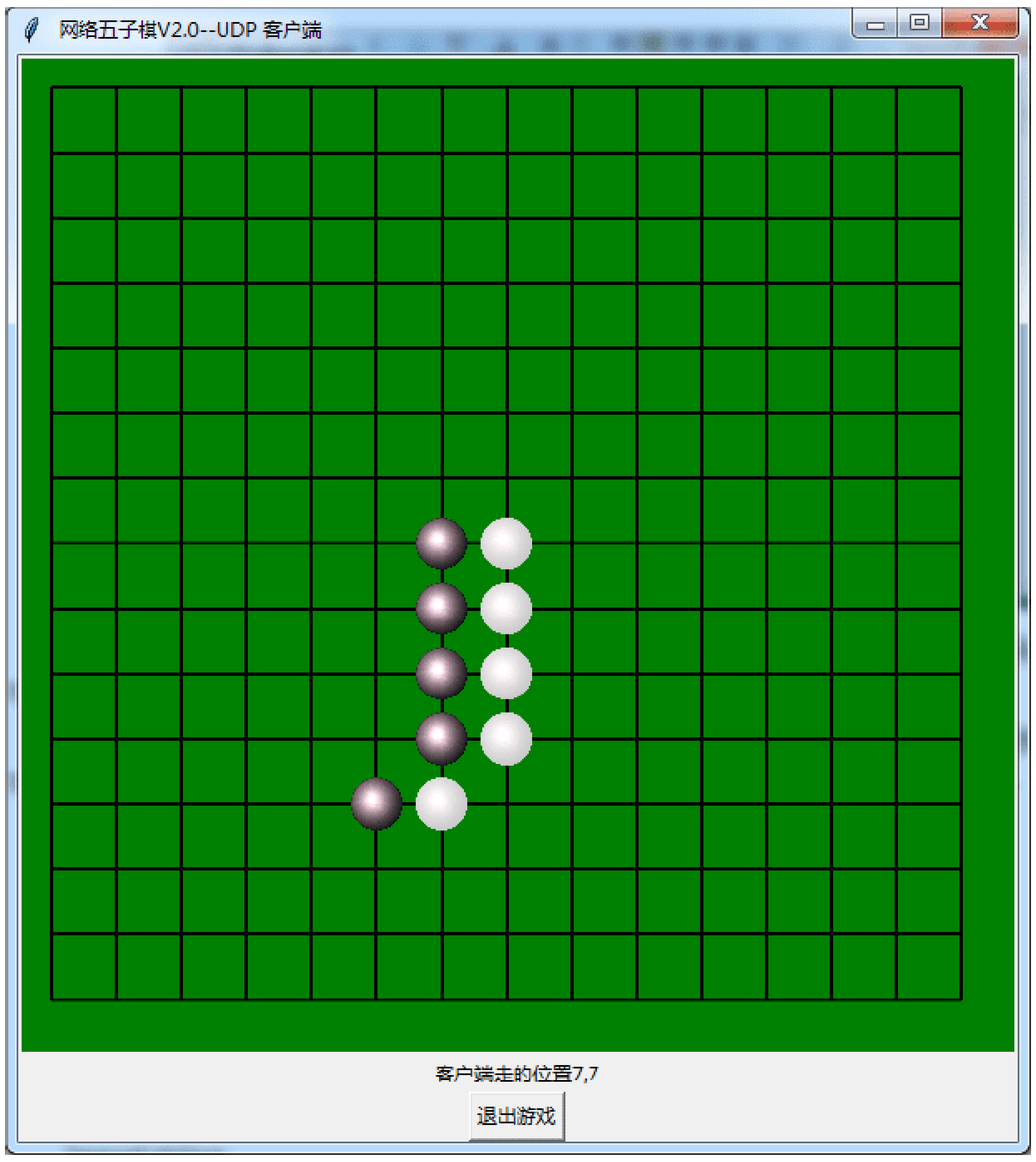


图 14-2　网络五子棋游戏的客户端界面

14.2　五子棋游戏的设计思想

在下棋过程中，为了保存所下过棋子的信息，使用列表 map。map[x][y]存储棋盘(x,y)处棋子的信息，如果为 0 代表黑子，为 1 代表白子。

游戏运行时，在鼠标单击事件中判断单击位置是否合法，即不能在已有棋子的位置单击，也不能超出游戏棋盘边界，如果合法，则将此位置信息加入到 map 列表和 back 列表（用于悔棋），同时调用 checkWin(x,y)判断游戏的输赢。

本游戏的设计关键是判断输赢的算法。对于该算法的具体实现，大致分为以下几个部分：

（1）判断 X=Y 轴上是否形成五子连珠。

（2）判断 X=−Y 轴上是否形成五子连珠。

（3）判断 Y 轴上是否形成五子连珠。

（4）判断 X 轴上是否形成五子连珠。

以上 4 种情况只要任何一种成立，那么就可以判断输赢。

```
def win_lose():                    #输赢判断
    #扫描整个棋盘，判断是否连成 5 颗
    a=str(turn)
    print("a=",a)
    for i in range(0,11):          #0～10
        #判断 X=Y 轴上是否形成五子连珠
        for j in range(0,11):      #0～10
            if map[i][j]==a and map[i+1][j+1]==a and map[i+2][j+2]==a and
            map[i+3][j+3]==a and map[i+4][j+4]==a:print("X=Y 轴上形成五子连珠")
                return True
    for i in range(4,15):          #4～14
        #判断 X=-Y 轴上是否形成五子连珠
        for j in range(0,11):      #0～10
            if map[i][j]==a and map[i-1][j+1]==a and map[i-2][j+2]==a and
            map[i-3][j+3]==a and map[i-4][j+4]==a:print("X=-Y 轴上形成五子连珠")
                return True
    for i in range(0,15):          #0～14
        #判断 Y 轴上是否形成五子连珠
        for j in range(4,15):      #4 ～14
            if map[i][j]==a and map[i][j-1]==a and map[i][j-2]==a and
            map[i][j-3]==a and map[i][j-4]==a:print("Y 轴上形成五子连珠")
                return True
    for i in range(0,11):          #0～10
        #判断 X 轴上是否形成五子连珠
        for j in range(0,15):      #0～14
            if map[i][j]==a and map[i+1][j]==a and map[i+2][j]==a
            and map[i+3][j]==a and map[i+4][j]==a:print("X 轴上形成五子连珠")
            return True
    return False
```

判断输赢实际上不用扫描整个棋盘，如果能得到刚下的棋子的位置(x, y)，就不用扫描整个棋盘，而仅仅在此棋子附近的横、竖、斜方向均判断一遍即可。

checkWin(x,y)判断这个棋子是否和其他棋子连成五子，即判断输赢。它是以(x,y)为中心通过横向、纵向、斜方向的判断来统计相同颜色的棋子个数。

例如以水平方向（横向）的判断为例，以(x, y)为中心计算水平方向上的棋子数量时首先向右最多 4 个位置，如果同色则 count 加 1，然后向左最多 4 个位置，如果同色则 count 加 1。统计完成后如果 count >=5，则说明水平方向连成五子。其他方向同理。因为每个方向判断前下子处(x,y)还有己方一个，所以 count 的初始值为 1。

```
def checkWin(x,y):
```

```
flag=False
count=1    #保存共有相同颜色的多少棋子相连
color=map[x][y]
#通过循环来做棋子相连的判断
#横向的判断
#判断横向是否有 5 个棋子相连，特点是纵坐标相同，即 map[x][y]中的 y 值是相同
i=1
while color==map[x+i][y] :  #向右统计
     count=count+1
     i=i+1
i=1
while color==map[x-i][y] :  #向左统计
     count=count+1
     i=i+1
if count>=5:
    flag=True
#纵向的判断
i2=1
count2=1
while color==map[x][y+i2]:
    count2=count2+1
    i2=i2+1
i2=1
while color==map[x][y-i2]:
    count2=count2+1
    i2=i2+1
if count2>=5:
    flag=True
#斜方向的判断（右上+左下）
i3=1
count3=1
while color==map[x+i3][y-i3]:
    count3=count3+1
    i3=i3+1
i3=1
while color==map[x-i3][y+i3]:
    count3=count3+1
    i3=i3+1
if count3>=5:
    flag=True
#斜方向的判断（右下+左上）
i4=1
```

```
        count4=1
        while color==map[x+i4][y+i4]:
            count4=count4+1
            i4=i4+1
        i4=1
        while color==map[x-i4][y-i4]:
            count4=count4+1
            i4=i4+1
        if count4>=5:
            flag=True
        return flag
```

在本程序中每下一步棋子，调用 checkWin(x,y) 函数判断是否已经连成五子，如果返回 True，则说明已经连成五子，显示输赢结果对话框。

14.3　关键技术

14.3.1　UDP 编程

视频讲解

TCP 是建立可靠连接，并且通信双方都可以用流的形式发送数据。相对于 TCP，UDP 则是面向无连接的协议。

在使用 UDP 协议时不需要建立连接，只需要知道对方的 IP 地址和端口号就可以直接发数据包，但是能不能到达就不知道了。虽然用 UDP 传输数据不可靠，但它的优点是和 TCP 相比速度快，对于不要求可靠到达的数据可以使用 UDP 协议。

通过 UDP 协议传输数据和 TCP 类似，使用 UDP 的通信双方也分为客户端和服务器端。图 14-3 所示为无连接 UDP 的时序图。

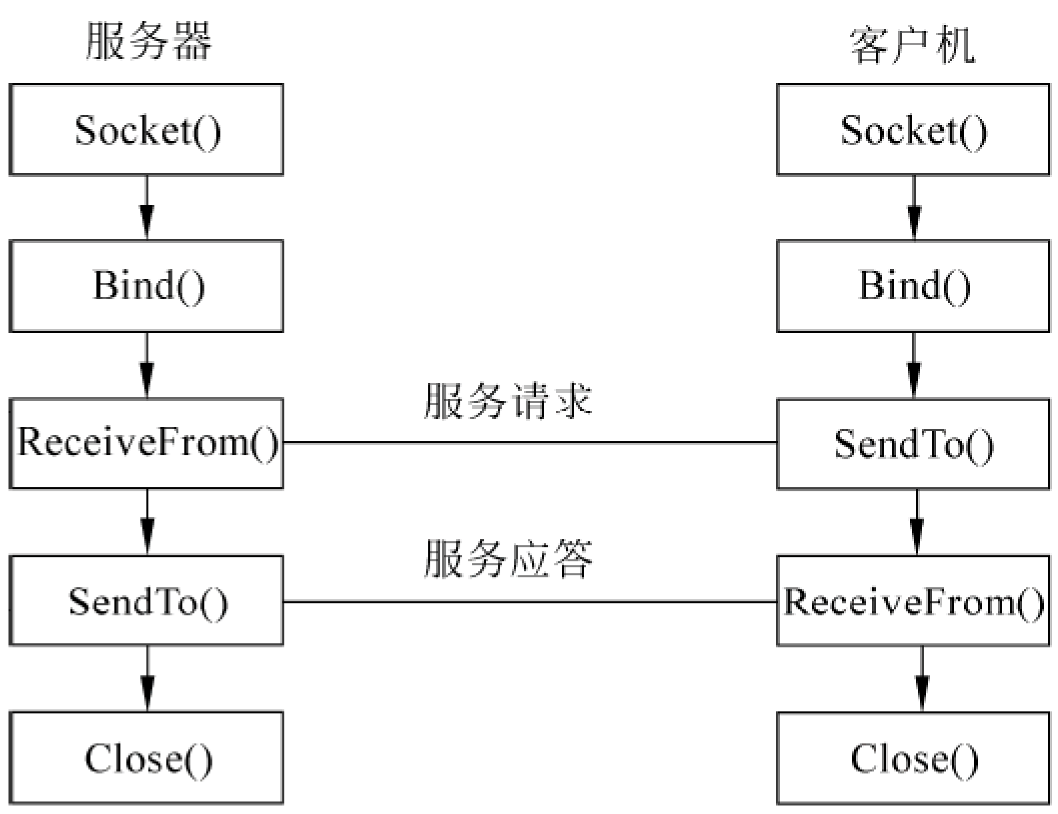


图 14-3　无连接 UDP 的时序图

对于 UDP C/S，客户机并不与服务器建立一个连接，而仅仅调用函数 SendTo()给服务器发送数据报。类似地，服务器也不从客户端接收一个连接，只是调用函数 ReceiveFrom()，等待从客户端来的数据。通常，依照 ReceiveFrom()得到的协议地址以及数据报，服务器就可以给客户送一个应答。

例 14-1 编写一个简单的 UDP 演示下棋程序。服务器端把 UDP 客户端发来的下棋坐标信息(x,y)显示出来，并把坐标加 1 后（模拟服务器端下棋）再发给 UDP 客户端。

服务器首先需要绑定 8888 端口：

```
import socket                                          #导入 socket 模块
s=socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.bind(('127.0.0.1', 8888))                            #绑定端口
```

在创建 Socket 时，SOCK_DGRAM 指定了这个 Socket 的类型是 UDP。绑定端口和 TCP 一样，但是不需要调用 listen()方法，而是直接接收来自任何客户端的数据：

```
print('Bind UDP on 8888...')
while True:
    #接收数据
    data, addr=s.recvfrom(1024)
    print('Received from %s:%s.' % addr)
    print('received:',data)
    p=data.decode('utf-8').split(",");  #decode()解码，将字节串转换成字符串
    x=int(p[0]);
    y=int(p[1]);
    print(p[0],p[1])
    pos=str(x+1)+","+str(y+1)           #模拟服务器端下棋的位置
    s.sendto(pos.encode('utf-8'),addr)  #发回客户端
```

recvfrom()方法返回数据和客户端的地址与端口，这样服务器收到数据后直接调用 sendto()就可以把数据用 UDP 发给客户端。

在客户端使用 UDP 时，首先创建基于 UDP 的 Socket，然后不需要调用 connect()，直接通过 sendto()给服务器发数据：

```
import socket                          #导入 socket 模块
s=socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
x=input("请输入 x 坐标")
y=input("请输入 y 坐标")
data=str(x)+","+str(y)
s.sendto(data.encode('utf-8'), ('127.0.0.1', 8888))
                                       #encode()编码，将字符串转换成传送的字节串
#接收服务器加 1 后的坐标数据
data2, addr=s.recvfrom(1024)
print("接收服务器加 1 后坐标数据: ", data2.decode('utf-8'))
                                       #decode()解码
s.close()
```

从服务器接收数据仍然调用 recvfrom()方法。

这里仍然用两个命令行分别启动服务器和客户端测试，运行效果如图 14-4 和图 14-5 所示。



图 14-4　服务器程序效果



图 14-5　客户端程序效果

上面模拟了服务器端和客户端两方下棋时的通信过程，有此基础可以实现基于 UDP 的网络五子棋游戏，真正开发出实用的网络程序。

14.3.2　自定义网络五子棋游戏的通信协议

网络五子棋游戏的设计难点在于需要与对方通信，这里使用了面向非连接的 Socket 编程。Socket 编程用于开发 C/S 结构程序，在这类应用中，客户端和服务器端通常需要先建立连接，然后发送和接收数据，交互完成后需要断开连接。本章的通信采用基于 UDP 的 Socket 编程实现。虽然这里两台计算机不分主次，但在设计时假设一台做服务器端（黑方），等待其他人加入。当其他人想加入的时候输入服务器端主机的 IP。为了区分通信中传送的是“输赢信息”“下的棋子位置信息”“结束游戏”等，在发送信息的首部加上标识。因此定义了如下协议：

❶ move|下的棋子位置坐标(x,y)

例如，“move|7,4”表示对方下子位置坐标(7,4)。

❷ over|哪方赢的信息

例如，“over |黑方你赢了”表示黑方赢了。

❸ exit|

表示对方离开了，游戏结束。

❹ join|

其表示连接服务器。

当然，可以根据程序功能增加协议，例如悔棋、文字聊天等协议。由于本程序没有设计“悔棋”和“文字聊天”功能，所以没定义相应的协议，读者可以自己完善程序。

在程序中根据接收的信息（当然都是字符串）通过字符串.split("|")获取消息类型（move、join、exit 或者 over），从中区分出“输赢信息 over”“下的棋子位置信息 move”等，代码如下：

```
def receiveMessage():                          #接收消息函数
    global s
    while True:
        #接收客户端发送的消息
        global addr
        data, addr=s.recvfrom(1024)
        data=data.decode('utf-8')
        a=data.split("|")                      #分割数据
        if not data:
            print('client has exited!')
            break
        elif a[0]=='join':                     #连接服务器请求
            print('client 连接服务器!')
            label1["text"]='client 连接服务器成功，请你走棋! '
        elif a[0]=='exit':                     #对方退出信息
            print('client 对方退出!')
            label1["text"]='client 对方退出，游戏结束! '
        elif a[0]=='over':                     #对方赢信息
            print('对方赢信息!')
            label1["text"]=data.split("|")[0]
            showinfo(title="提示",message=data.split("|")[1])
        elif a[0]=='move':                     #客户端走的位置信息，例如"move|7,4"
            print('received:',data,'from',addr)
            p=a[1].split(",")
            x=int(p[0]);
            y=int(p[1]);
            print(p[0],p[1])
            label1["text"]="客户端走的位置"+p[0]+p[1]
            drawOtherChess(x,y)                #画对方棋子
    s.close()
```

掌握通信协议以及单机版五子棋游戏的知识之后就可以开发网络五子棋游戏了。

14.4　网络五子棋游戏程序设计的步骤

视频讲解

14.4.1　服务器端程序设计的步骤

❶ 主程序

定义含两个棋子图片的列表 imgs，创建 Window 窗口对象 root，初始化游戏地图 map，绘制 15×15 的游戏棋盘，添加显示提示信息的标签 Label，绑定 Canvas 的鼠标和按钮左键单击事件。

同时创建 UDP 通信服务器端的 SOCKET，绑定在 8000 端口，启动线程接收客户端的消息，最后的 root.mainloop()方法是进入窗口的主循环，也就是显示窗口。

```
from tkinter import *
from tkinter.messagebox import *
import socket
import threading
import os

root=Tk()
root.title("网络五子棋 V2.0—服务器端")
#五子棋——夏敏捷 2016-2-11
imgs=[PhotoImage(file='D:\\python\\bmp\\BlackStone.gif'),
             PhotoImage(file='D:\\python\\bmp\\WhiteStone.gif')]
turn=0                         #轮到哪方走棋，0 是黑方，1 是白方
Myturn=-1                      #保存自己的角色，-1 表示还没确定下来
map=[[" "," "," "," "," "," "," "," "," "," "," "," "," "," "," "]for y
in range(15)]
cv=Canvas(root, bg='green', width=610, height=610)
drawQiPan()                                #绘制 15×15 的游戏棋盘
cv.bind("<Button-1>", callpos)
cv.pack()
label1=Label(root,text="服务器端...")       #显示提示信息
label1.pack()
button1=Button(root,text="退出游戏")        #按钮
button1.bind("<Button-1>", callexit)
button1.pack()
#创建 UDP SOCKET
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.bind(('localhost',8000))
addr=('localhost',8000)
startNewThread()                           #启动线程接收客户端的消息
```

```
    receiveMessage();
    root.mainloop()
```

❷ 退出函数

“退出游戏”按钮单击事件的代码很简单，仅仅发送一个"exit|"命令协议消息，最后调用 os._exit(0)结束程序。

```
def callexit(event):              #退出
    pos="exit|"
    sendMessage(pos)
    os._exit(0)
```

❸ 走棋函数

在鼠标单击事件中完成走棋功能，判断单击位置是否合法，既不能在已有棋的位置单击，也不能超出游戏棋盘边界，如果合法，则将此位置信息记录到 map 列表（数组）中。

同时由于是网络对战，第一次走棋时还要确定自己的角色（是白方还是黑方），而且要判断是否轮到自己走棋。这里使用两个变量 Myturn、turn 来解决。

```
Myturn=-1                         #保存自己的角色
```

Myturn 是−1 表示还没确定下来，在第一次走棋时修改。

turn 保存轮到谁走棋，如果 turn 是 0 表示轮到黑方，turn 是 1 表示轮到白方。

最后是本游戏的关键——输赢判断。在程序中调用 win_lose()函数判断输赢，判断在 4 种情况下是否连成五子，返回 True 或 False。根据当前走棋方 turn 的值（0 为黑方，1 为白方）得出谁赢。

自己走完后轮到对方走棋。

```
def callpos(event):               #走棋
    global turn
    global Myturn
    if Myturn==-1:                #第一次确定自己的角色（是白方还是黑方）
        Myturn=turn
    else:
        if(Myturn!=turn):
            showinfo(title="提示",message="还没轮到自己走棋")
            return
    #print("clicked at", event.x, event.y, turn)
    x=(event.x)//40               #换算棋盘坐标
    y=(event.y)//40
    print("clicked at", x, y, turn)
    if map[x][y]!=" ":
        showinfo(title="提示",message="已有棋子")
    else:
        img1=imgs[turn]
        cv.create_image((x*40+20,y*40+20),image=img1)    #画自己的棋子
        cv.pack()
```

```
            map[x][y]=str(turn)
            pos=str(x)+","+str(y)
            sendMessage("move|"+pos)
            print("服务器走的位置",pos)
            label1["text"]="服务器走的位置"+pos
            #输出输赢信息
            if win lose()==True:
                if turn==0:
                    showinfo(title="提示",message="黑方你赢了")
                    sendMessage("over|黑方你赢了")
                else:
                    showinfo(title="提示",message="白方你赢了")
                    sendMessage("over|白方你赢了")
            #换下一方走棋
            if turn==0:
                turn=1
            else:
                turn=0
```

❹ 画对方棋子

当轮到对方走棋子后，在自己的棋盘上根据 turn 知道对方角色，从 socket 获取对方走棋坐标(x,y)，从而画出对方棋子。在画出对方棋子后，同样换下一方走棋。

```
def drawOtherChess(x,y):                #画对方棋子
        global turn
        img1=imgs[turn]
        cv.create_image((x*40+20,y*40+20),image=img1)
        cv.pack()
        map[x][y]=str(turn)
        #换下一方走棋
        if turn==0:
            turn=1
        else:
            turn=0
```

❺ 画棋盘

drawQiPan()画 15×15 的五子棋棋盘。

```
def drawQiPan():                        #画棋盘
    for i in range(0,15):
        cv.create_line(20,20+40*i,580,20+40*i,width=2)
    for i in range(0,15):
        cv.create_line(20+40*i,20,20+40*i,580,width=2)
    cv.pack()
```

❻ 输赢判断

win_lose()从 4 个方向扫描整个棋盘，判断是否连成 5 子。

```
def win lose():          #输赢判断
    #扫描整个棋盘，判断是否连成 5 子
    a=str(turn)
    print("a=",a)
    for i in range(0,11):       #0～10
        #判断 X=Y 轴上是否形成五子连珠
        for j in range(0,11):   #0～10
            if map[i][j]==a and map[i+1][j+1]==a and map[i+2][j+2]==a and
            map[i+3][j+3]==a and map[i+4][j+4]==a:print("X=Y 轴上形成五子连珠")
                return True
    for i in range(4,15):       #4～14
        #判断 X=-Y 轴上是否形成五子连珠
        for j in range(0,11):   #0～10
            if map[i][j]==a and map[i-1][j+1]==a and map[i-2][j+2]==a and
            map[i-3][j+3]==a and map[i-4][j+4]==a:print("X=-Y 轴上形成五子连珠")
                return True
    for i in range(0,15):       #0～14
        #判断 Y 轴上是否形成五子连珠
        for j in range(4,15):   #4～14
            if map[i][j]==a and map[i][j-1]==a and map[i][j-2]==a and
            map[i][j-3]==a and map[i][j-4]==a:print("Y 轴上形成五子连珠")
                return True
    for i in range(0,11):       #0～10
        #判断 X 轴上是否形成五子连珠
        for j in range(0,15):   #0～14
            if map[i][j]==a and map[i+1][j]==a and map[i+2][j]==a and
            map[i+3][j]==a and map[i+4][j]==a:print("X 轴上形成五子连珠")
                return True
    return False
```

❼ 输出 map 地图

这里主要是显示当前棋子信息。

```
def print map():                  #输出 map 地图
    for j in range(0,15):         #0～14
      for i in range(0,15):       #0～14
         print(map[i][j],end=' ')
      print('w')
```

❽ 接收消息

本程序的关键部分就是接收消息 data，从 data 字符串.split("|")中分割出消息类型

(move、join、exit 或者 over)。如果为'join'，是客户端连接服务器请求；如果为'exit'，是对方客户端退出信息；如果为' move '，是客户端走的位置信息；如果为' over '，是对方客户端赢的信息。这里重点处理对方的走棋信息，例如"move|7,4"，通过字符串.split(",")分割出(x,y)坐标。

```
def receiveMessage():
    global s
    while True:
        #接收客户端发送的消息
        global addr
        data, addr=s.recvfrom(1024)
        data=data.decode('utf-8')
        a=data.split("|")                    #分割数据
        if not data:
            print('client has exited!')
            break
        elif a[0]=='join':                   #连接服务器请求
            print('client 连接服务器!')
            label1["text"]='client 连接服务器成功，请你走棋！'
        elif a[0]=='exit':                   #对方退出信息
            print('client 对方退出!')
            label1["text"]='client 对方退出，游戏结束！'
        elif a[0]=='over':                   #对方赢信息
            print('对方赢信息!')
            label1["text"]=data.split("|")[0]
            showinfo(title="提示",message=data.split("|")[1])
        elif a[0]=='move':                   #客户端走的位置信息"move|7,4"
            print('received:',data,'from',addr)
            p=a[1].split(",")
            x=int(p[0]);
            y=int(p[1]);
            print(p[0],p[1])
            label1["text"]="客户端走的位置"+p[0]+p[1]
            drawOtherChess(x,y)              #画对方棋子
    s.close()
```

❾ 发送消息

发送消息的代码很简单，调用 Socket 的 sendto()函数就可以把按协议写的字符串信息发出。

```
def sendMessage(pos):                #发送消息
    global s
    global addr
```

```
    s.sendto(pos.encode(),addr)
```

❿ 启动线程接收客户端的消息

```
#启动线程接收客户端的消息
def startNewThread():
       #启动一个新线程来接收客户器端的消息
       #thread.start new thread(function,args[,kwargs])
       #其中，function 参数是将要调用的线程函数，args 是传递给线程函数的参数，它必须
       #是元组类型，而 kwargs 是可选的参数
       #receiveMessage 函数不需要参数，只传一个空元组
       thread=threading.Thread(target=receiveMessage,args=())
       thread.setDaemon(True);
       thread.start();
```

至此完成服务器端程序设计。图 14-6 所示为服务器端走棋过程中打印的输出信息。网络五子棋的客户端程序设计基本上与服务器端相似，主要区别在消息处理上。

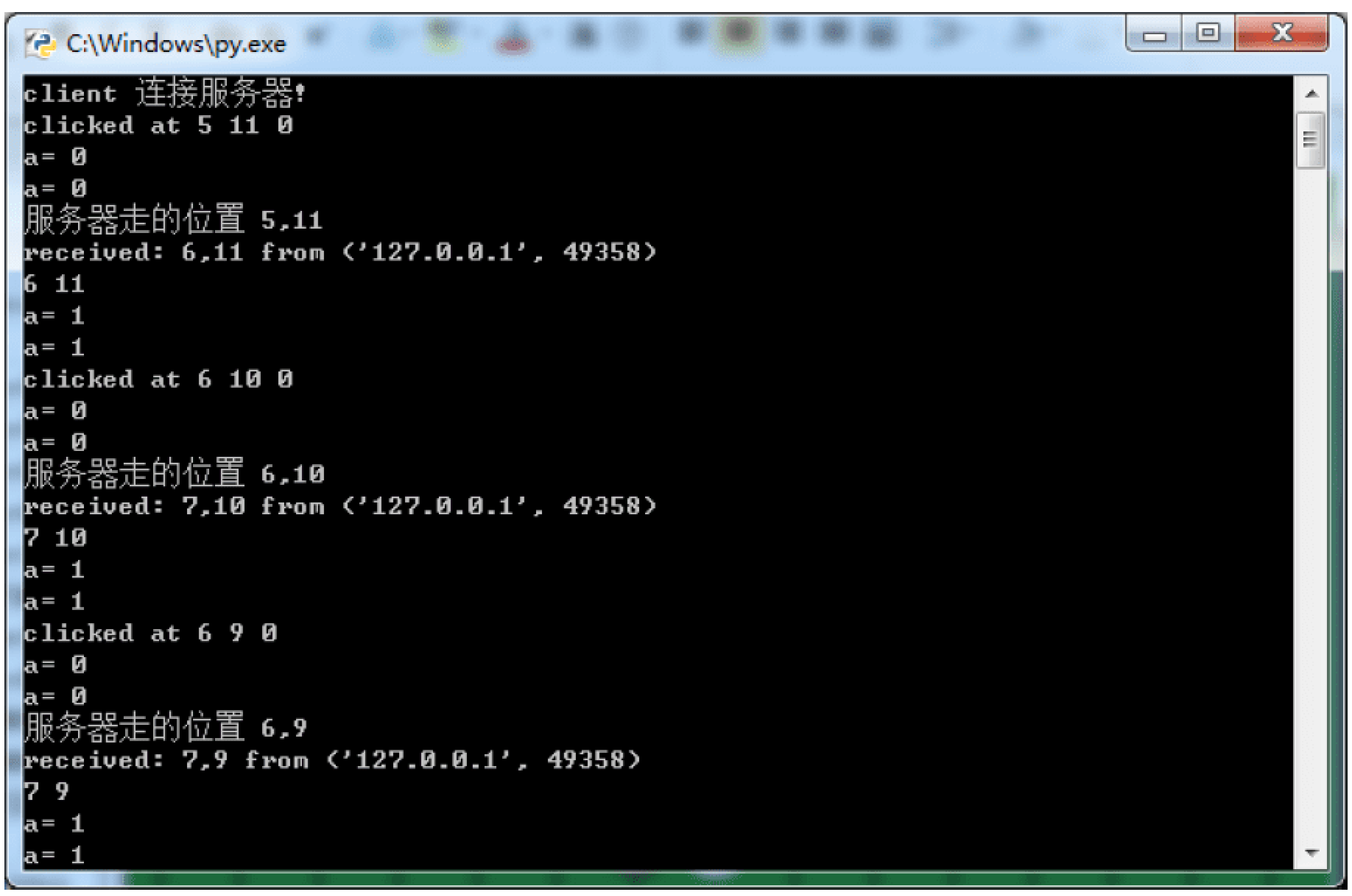


图 14-6　走棋过程中打印的输出信息

14.4.2　客户端程序设计的步骤

❶ 主程序

定义含两个棋子图片的列表 imgs，创建 Window 窗口对象 root，初始化游戏地图 map，绘制 15×15 的游戏棋盘，添加显示提示信息的标签 Label，绑定 Canvas 的鼠标和按钮左键单击事件。

同时创建 UDP 通信客户端的 SOCKET，这里不指定端口，会自动绑定某个空闲端口，

由于是客户端 SOCKET，需要指定服务器端的 IP 和端口号，并发出连接服务器端请求。

启动线程接收服务器端的消息，最后的 root.mainloop()方法是进入窗口的主循环，也就是显示窗口。

```
from tkinter import *
from tkinter.messagebox import *
import socket
import threading
import os
root=Tk()
root.title("网络五子棋 V2.0—UDP 客户端")
imgs=[PhotoImage(file='D:\\python\\bmp\\BlackStone.gif'),
PhotoImage(file='D:\\python\\bmp\\WhiteStone.gif')]
turn=0
Myturn=-1
map=[[" "," "," "," "," "," "," "," "," "," "," "," "," "," "," "]for y in
range(15)]
cv=Canvas(root, bg='green', width = 610, height = 610)
drawQiPan()
cv.bind("<Button-1>", callback)
cv.pack()
label1=Label(root,text="客户端...")
label1.pack()
button1=Button(root,text="退出游戏")
button1.bind("<Button-1>", callexit)
button1.pack()
#创建 UDP SOCKET
s=socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
port=8000                    #服务器端口
host='localhost'             #服务器地址 192.168.0.101
pos='join|'                  #“连接服务器”命令
sendMessage(pos);            #发送连接服务器请求
startNewThread()             #启动线程接收服务器端的消息
receiveMessage();
root.mainloop()
```

❷ 退出函数

“退出游戏”按钮单击事件的代码很简单，仅仅发送一个"exit|"命令协议消息，最后调用 os._exit(0)结束程序。

```
def callexit(event):    #退出
    pos="exit|"
```

```
    sendMessage(pos)
    os._exit(0)
```

❸ 走棋函数

其功能与服务器端相同，仅仅是提示信息不同。

```
def callback(event):          #走棋
    global turn
    global Myturn
    if Myturn==-1:            #第一次确定自己的角色（是白方还是黑方）
        Myturn=turn
    else:
        if(Myturn!=turn):
            showinfo(title="提示",message="还没轮到自己走棋")
            return
    #print("clicked at", event.x, event.y, turn)
    x=(event.x)//40           #换算棋盘坐标
    y=(event.y)//40
    print("clicked at", x, y, turn)
    if map[x][y]!=" ":
        showinfo(title="提示",message="已有棋子")
    else:
        img1=imgs[turn]
        cv.create_image((x*40+20,y*40+20),image=img1)
        cv.pack()
        map[x][y]=str(turn)
        pos=str(x)+","+str(y)
        sendMessage("move|"+pos)
        print("客户端走的位置",pos)
        label1["text"]="客户端走的位置"+pos
        #输出输赢信息
        if win_lose()==True:
            if turn==0:
                showinfo(title="提示",message="黑方你赢了")
                sendMessage("over|黑方你赢了")
            else:
                showinfo(title="提示",message="白方你赢了")
                sendMessage("over|白方你赢了")
        #换下一方走棋
        if turn==0:
            turn=1
        else:
            turn=0
```

❹ 画棋盘

drawQiPan()画 15×15 的五子棋棋盘。

```
def drawQiPan():          #画棋盘
    for i in range(0,15):
        cv.create_line(20,20+40*i,580,20+40*i,width=2)
    for i in range(0,15):
        cv.create_line(20+40*i,20,20+40*i,580,width=2)
    cv.pack()
```

❺ 输赢判断

win_lose()从 4 个方向扫描整个棋盘，判断是否连成五子。其功能与服务器端相同，代码没有区别，因此将代码省略了。

❻ 接收消息

接收消息 data，从 data 字符串.split("|")中分割出消息类型（move、join、exit 或者 over）。其功能与服务器端没有区别，仅仅是没有'join'消息类型，因为客户端连接服务器，而服务器不会连接客户端，所以少了一个'join'消息类型判断。

```
def receiveMessage():                              #接收消息
    global s
    while True:
        data=s.recv(1024).decode('utf-8')
        a=data.split("|")                          #分割数据
        if not data:
            print('server has exited!')
            break
        elif a[0]=='exit':                         #对方退出信息
            print('对方退出!')
            label1["text"]='对方退出，游戏结束! '
        elif a[0]=='over':                         #对方赢信息
            print('对方赢信息!')
            label1["text"]=data.split("|")[0]
            showinfo(title="提示",message=data.split("|")[1])
        elif a[0]=='move':                         #服务器端走的位置信息
            print('received:',data)
            p=a[1].split(",")
            x=int(p[0]);
            y=int(p[1]);
            print(p[0],p[1])
            label1["text"]="服务器端走的位置"+p[0]+p[1]
            drawOtherChess(x,y)                    #画对方棋子，函数代码同服务器端
    s.close()
```

❼ 发送消息

发送消息的代码很简单，仅仅调用 Socket 的 sendto()函数，就可以把按协议写的字符

串信息发出。

```
def sendMessage(pos):                              #发送消息
    global s
    s.sendto(pos.encode(),(host,port))
```

❽ **启动线程接收服务器端的消息**

```
#启动线程接收服务器端的消息
def startNewThread():
        #启动一个新线程来接收服务器端的消息
        #thread.start new thread(function,args[,kwargs])
        #其中，function 参数是将要调用的线程函数，args 是传递给线程函数的参数，它必须
        #是元组类型，而 kwargs 是可选的参数
        #receiveMessage 函数不需要参数，只传一个空元组
        thread=threading.Thread(target=receiveMessage,args=())
        thread.setDaemon(True);
        thread.start();
```

至此完成客户端程序设计。

第 15 章

益智游戏——中国象棋

中国象棋和五子棋一样，也是一种家喻户晓的棋类游戏，它的多变吸引了无数玩家。在信息化的今天，再用纸棋盘、木棋子下象棋有点太落伍，能否来点革新，把古老的象棋请进计算机呢？本章介绍制作“中国象棋”游戏的原理和过程。

15.1 中国象棋介绍

视频讲解

❶ 棋盘

棋子活动的场所叫作“棋盘”，在长方形的平面上，9 条平行的竖线和 10 条平行的横线相交，共 90 个交叉点，棋子就摆在这些交叉点上。其中的第 5、第 6 两横线之间未画竖线的空白地带称为“河界”，整个棋盘以“河界”分为相等的两部分；两方将帅坐镇、画有“米”字方格的地方叫作“九宫”。

❷ 棋子

象棋的棋子共 32 个，分为红、黑两组，各 16 个，由对弈双方各执一组，每组兵种是一样的，各分为 7 种。

- 红方：帅、仕、相、车、马、炮、兵。
- 黑方：将、士、象、车、马、炮、卒。

其中，帅与将、仕与士、相与象、兵与卒的作用完全相同，仅仅是为了区分红棋和黑棋。

❸ 各棋子的走法说明

1）帅与将

移动范围：只能在王宫内移动。

移动规则：每一步只可以水平或垂直移动一点。

2）仕与士

移动范围：只能在王宫内移动。

移动规则：每一步只可以沿对角线方向移动一点。

3）相与象

移动范围：河界的一侧。

移动规则：每一步只可以沿对角线方向移动两点，另外，在移动的过程中不能够穿越障碍。

4）马

移动范围：任何位置。

移动规则：每一步只可以水平或垂直移动一点，再按对角线方向向左或者向右斜着移动一点，俗称“马走日”。另外，在移动的过程中不能够穿越障碍。

5）车

移动范围：任何位置。

移动规则：可以在水平或垂直方向移动任意个无阻碍的点。

6）炮

移动范围：任何位置。

移动规则：移动起来和车很相似，但它必须跳过一个棋子去吃掉对方的一个棋子。

7）兵与卒

移动范围：任何位置。

移动规则：每一步只能向前移动一点。过河以后，它便增加了向左/右移动的能力，兵不允许向后移动。

❹ 关于胜、负、和

在对局中，若出现下列情况之一，本方输，对方赢：

（1）己方的帅（将）被对方棋子吃掉。

（2）己方发出认输请求。

（3）己方走棋超出步时限制。

15.2　关键技术

❶ 移动指定图形对象

使用 move()方法可以修改图形对象（例如一个棋子）的坐标，具体语法如下：

```
Canvas 对象.move(图形对象,x 坐标偏移量,y 坐标偏移量)
```

例如移动“帅”棋子图片向右 150 像素、向下 150 像素，从矩形左上角移到右下角。

```
from tkinter import *
def callback():                           #事件处理函数
    cv.move(rt1,150,150)                  #移动 rt1
root=Tk()
root.title('移动"帅"棋子')                  #设置窗口标题
#创建一个 Canvas，设置其背景色为白色
cv=Canvas(root, bg='white', width=260, height=220)
img1=PhotoImage(file='红帅.png')
```

```
cv.create_rectangle(40,40,190,190,outline='red',fill='green')
rt1=cv.create_image((40,40),image=img1)         #绘制"帅"棋子图片
cv.pack()
button1=Button(root,text="移动棋子",command=callback,fg="red")
button1.pack()
root.mainloop()
```

为了对比移动图形对象的效果，程序在(40,40,190,190)位置绘制了一个矩形（由绿色填充），单击“移动棋子”按钮后，“帅”棋子 rt1 通过 move()方法移动到矩形右下角，出现如图 15-1 所示的效果。

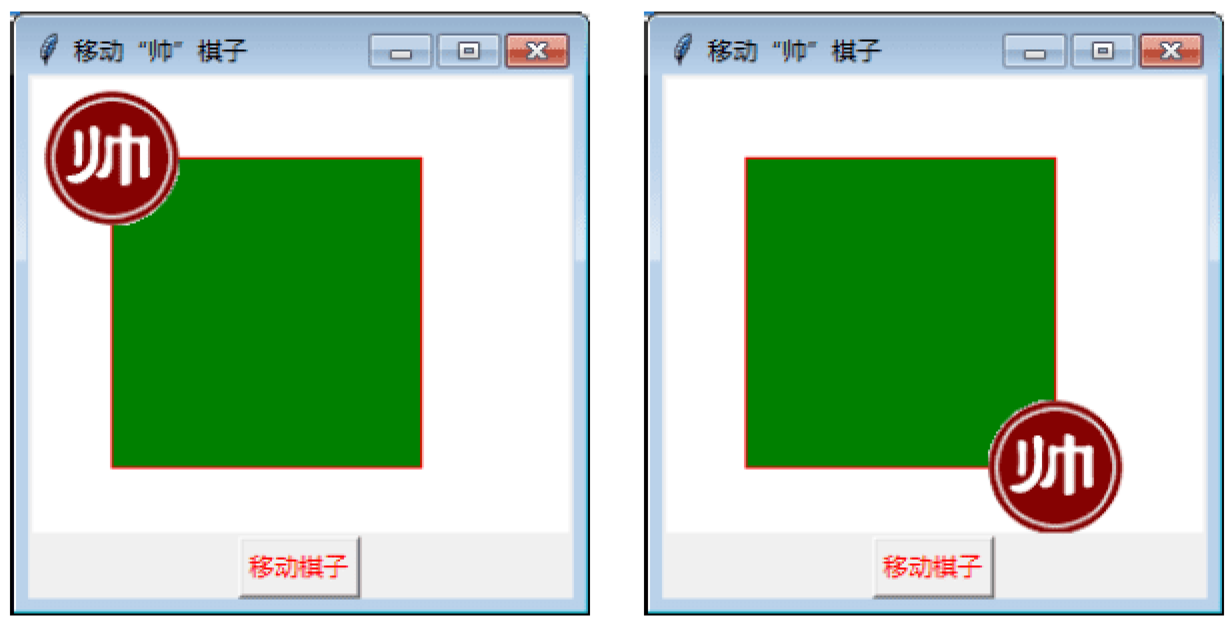


图 15-1　移动“帅”棋子图形对象

❷ 删除指定图形对象

使用 delete()方法可以删除图形对象（例如选中棋子的提示框），具体语法如下：

```
Canvas 对象.delete(图形对象)
```

将前面例子中的最后一行改成如下 5 行：

```
def callback2():                #事件处理函数
    cv.delete(rt1)              #删除 rt1
button2=Button(root,text="删除棋子",command=callback2,fg="red")
button2.pack()
root.mainloop()
```

单击“删除棋子”按钮，“帅”棋子消失，出现如图 15-2 所示的效果。

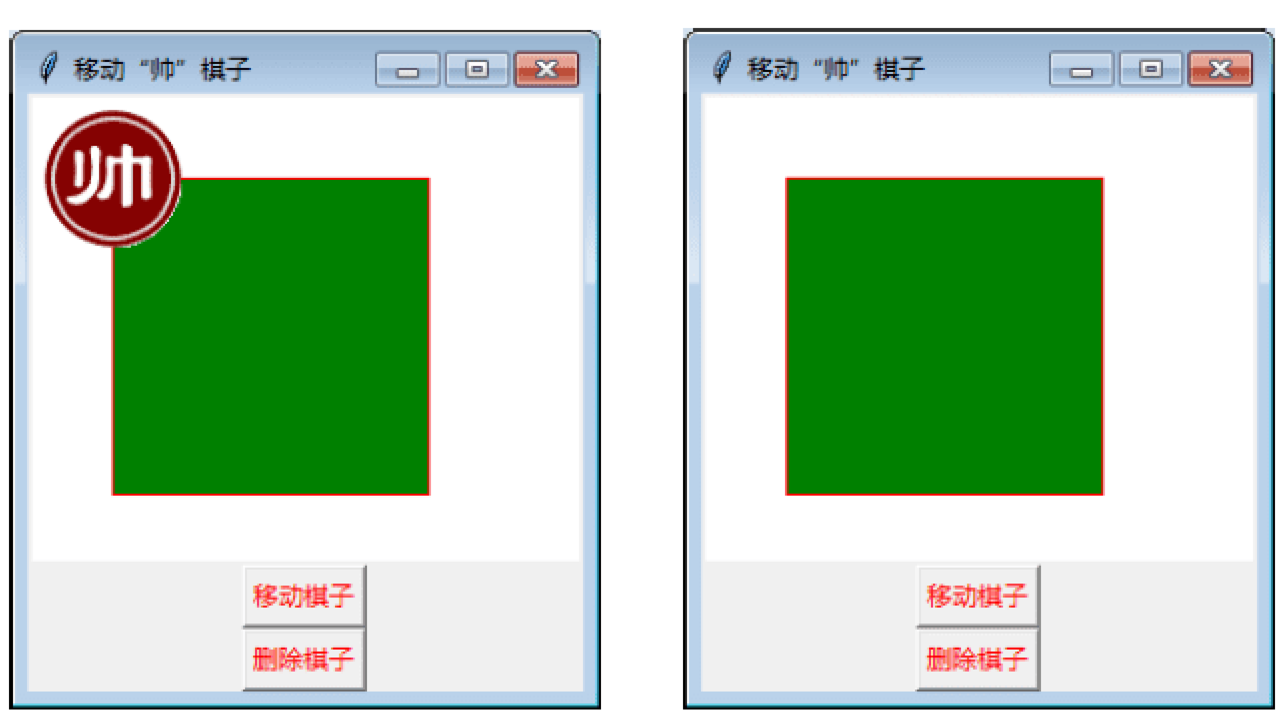


图 15-2　删除指定图形对象

15.3 中国象棋的设计思路

❶ 棋盘的表示

棋盘的表示就是使用一种数据结构来描述棋盘及棋盘上的棋子，这里使用一个二维列表 chessmap 来表示。一个典型的中国象棋棋盘使用 9×10 的二维列表（数组）来表示，每一个元素代表棋盘上的一个交点，一个没有棋子的交点所对应的元素是−1。一个二维列表（数组）chessmap 保存了当前棋盘的布局。当 chessmap[x][y]=i 时说明(x,y)处是棋子图像 i，否则 chessmap[x][y]=−1，表示此处为空（无棋子）。

该程序中下棋的棋盘界面通过 DrawBoard()函数在 Canvas 对象 cv 上画出。

```
img1=PhotoImage(file='D:\\python\\bmp\\棋盘.png')
def DrawBoard():                    #画棋盘
    p1=cv.create_image((0,0),image=img1)
    cv.coords(p1,(360,400))         #指定棋盘图像的中心点坐标为(360,400)
```

❷ 棋子的表示

棋子的显示需要图片，每种棋子的图案和棋盘使用的对应图片资源如图 15-3 所示。该游戏中红方在南，黑方在北。



图 15-3　棋子图片资源

❸ 走棋规则

对于象棋来说，有马走“日”、象走“田”等一系列复杂的规则。走法的产生是博弈程序中一个相当复杂而且耗费运算时间的工作，不过，通过良好的数据结构可以显著地提高生成的速度。

判断是否能走棋的算法如下：

根据棋子名称的不同，按相应规则进行判断。

（1）如果为“车”，检查是否走直线，及中间是否有子。

（2）如果为“马”，检查是否走“日”字，是否蹩脚。

（3）如果为“炮”，检查是否走直线，判断是否吃子，如果吃子，则检查中间是否只有一个棋子；如果不吃，则检查中间是否有棋子。

（4）如果为“兵”或“卒”，检查是否走直线，走一步及向前走，根据是否过河，检查是否横走。

（5）如果为“将”或“帅”，检查是否走直线，走一步及是否超过范围。

（6）如果为“士”或“仕”，检查是否走斜线，走一步及是否超出范围。

（7）如果为“象”或“相”，检查是否走“田”字，是否蹩脚，及是否超出范围。

那么如何分辨棋子？在程序中采用了棋子图形对象来获取。

在该程序中，IsAbleToPut(id,x,y,oldx,oldy)函数实现判断是否能走棋并返回逻辑值，它的代码较复杂，其参数含义如下：

参数 id 代表走的棋子图形对象；因为 dict_ChessName 字典中存储的是 id 对应的棋子名（例如"红马"），如果 qi_name = dict_ChessName[id]，获取棋子名含颜色信息，而字符串[1]可以获取字符串的第 2 个字符，所以 dict_ChessName[id][1]意味着取字符串的第 2 个字符，例如"红马"取第 2 个字符得到"马"。

参数 x 和 y 代表走棋的目标位置。参数 oldx 和 oldy 代表走动棋子的原始位置。

IsAbleToPut(id, x, y,oldx,oldy)函数实现走棋规则的判断。

例如“将”或“帅”的走棋规则：只能走一格，所以原 x 坐标与新位置 x 坐标之差不能大于 1，原 y 坐标与新位置 y 坐标之差不能大于 1。

```
if(abs(x-oldx)>1 or abs(y-oldy)>1):
    return False;
```

由于不能走出九宫，所以 x 坐标为 3、4、5 且 0≤y≤2 或 7≤y≤9（因为走棋时自己的“将”或“帅”只能在九宫中），否则此步违规，将返回 False。

```
if(x<3 or x>5 or (y>=3 and y<=6)):
    return False;
```

最终“将”或“帅”的走棋规则如下：

```
#“将”或“帅”的走棋判断
if(qi_name=="将" or qi_name=="帅"):
    if((x-oldx)*(y-oldy)!=0):       #斜线走棋
        return False;
    if(abs(x-oldx)>1 or abs(y-oldy)>1):
        return False;
    if(x<3 or x>5 or (y>=3 and y<=6)):
        return False;
    return True;
```

“仕”或“士”的走棋规则：只能走斜线一格，所以原 x 坐标与新位置 x 坐标之差为 1，且原 y 坐标与新位置 y 坐标之差也为 1。

```
    if(qi_name=="士" or qi_name=="仕"):
        if((x-oldx)*(y-oldy)==0):
            return False;
        if(abs(x-oldx)>1 or abs(y-oldy)>1):
            return False;
```

由于不能走出九宫，所以 x 坐标为 3、4、5 且 0≤y≤2 或 7≤y≤9，否则此步违规，将返回 False。

```
    if(x<3 or x>5 or (y>=3 and y<=6)):
        return False;
```

“炮”的走棋规则：只能走直线，所以 x 和 y 不能同时改变，即(x − oldx) * (y − oldy) = 0 保证走直线。然后判断如果 x 坐标改变了，原位置 oldx 到目标位置 x 之间是否有棋子，如果有，则累加之间棋的个数 c。通过 c 是否为 1 且目标处非己方棋子，可以判断是否可以走棋。同样的方法判断“炮”的 y 坐标改变时是否可以走棋。

“兵”或“卒”的走棋规则：只能向前走一步，根据是否过河检查是否横走，所以 x 与原坐标 oldx 改变的值不能大于 1，同时 y 与原坐标 oldy 改变的值也不能大于 1。例如红兵如果过河即是 y<5（游戏时红方在南）。

```
    #“卒”或“兵”的走棋判断
    if(qi_name=="卒" or qi_name=="兵"):               #红方在南，黑方在北
        if((x-oldx) * (y-oldy)!=0):                   #不是直线走棋
            return False;
        if(abs(x-oldx)>1 or abs(y-oldy)>1):           #走多步，不符合兵仅能走一步
            return False;
        if(y>=5 and (x-oldx)!=0 and qi_name=="兵"):
                                                      #红兵未过河且横向走棋
            return False;
        if(y<5 and (x-oldx)!=0 and qi_name=="卒"):    #黑卒未过河且横向走棋
            return False;
        if(y-oldy>0 and qi_name=="兵"):               #兵后退
            return False;
        if(y-oldy<0 and qi_name=="卒"):               #卒后退
            return False;
        return True;
```

其余棋子的判断方法类似，这里不再一一介绍。

❹ 坐标的转换

整个棋盘左上角的坐标为(0,0)，右下角的坐标为(8,9)，如图 15-4 所示。例如“黑车”的初始位置即为(0,0)，“黑将”的初始位置即为(4,0)，“红帅”的初始位置即为(4,9)。在走棋过程中，需要将鼠标像素坐标转换成棋盘坐标，棋盘方格的大小是 76 像素，通过整除 76 解析出棋盘坐标(x,y)。

```
        x=(event.x-14)//76          #换算成棋盘坐标
        y=(event.y-14)//76
```

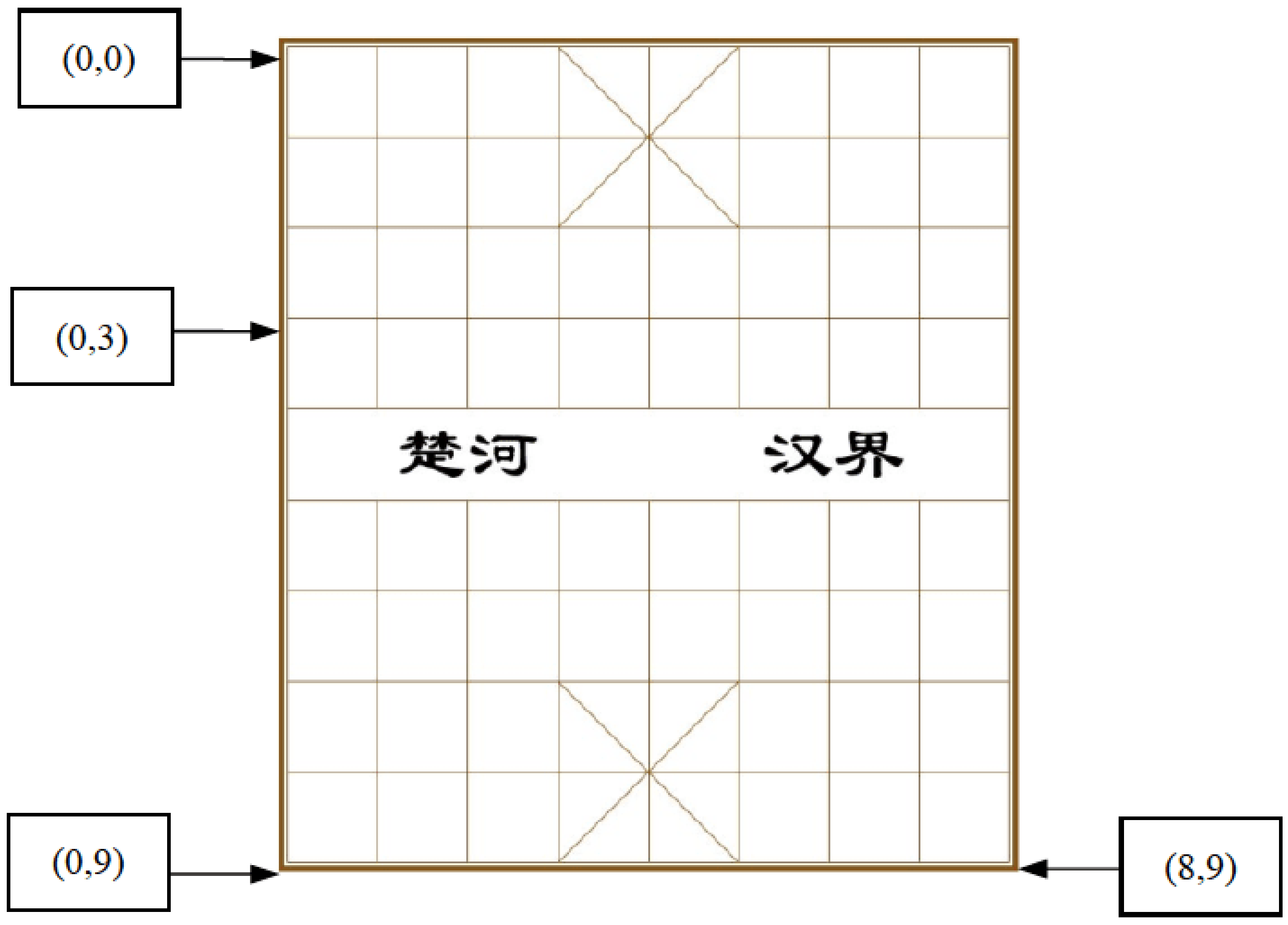


图 15-4　棋盘坐标示意图

15.4　中国象棋实现的步骤

视频讲解

首先导入 tkinter 库。

```
from tkinter import *
from tkinter.messagebox import *
```

创建一个 Canvas，设置其背景色为白色，用 Canvas 显示棋盘中所有的棋子。imgs 是 PhotoImage 对象列表，获取所有的棋子图片。

```
dict_ChessName={}        #定义一个字典
```

例如，本游戏中字典 dict_ChessName 存储的内容如下：

{2: '黑车', 3: '黑马', 4: '黑象', 5: '黑士', 6: '黑将', 7: '黑士', 8: '黑象', 9: '黑马', 10: '黑车', 11: '黑卒', 12: '黑卒', 13: '黑卒', 14: '黑卒', 15: '黑卒', 16: '黑炮', 17: '黑炮', 18: '红车', 19: '红马', 20: '红相', 21: '红仕', 22: '红帅', 23: '红仕', 24: '红相', 25: '红马', 26: '红车', 27: '红兵', 28: '红兵', 29: '红兵', 30: '红兵', 31: '红兵', 32: '红炮', 33: '红炮'}

字典的 Key 为每个棋子图像的 id，Value 是棋子种类名，例如图像对象 11 对应的是黑卒。因为首先建立 Canvas 对象 id=0 和棋盘对象 id=1，所以棋子图像的 id 是从 2 开始的。

```
root=Tk()
#创建一个 Canvas，设置其背景色为白色
cv=Canvas(root, bg='white', width=720, height=800)
chessname=["黑车","黑马","黑象","黑士","黑将","黑士","黑象","黑马","黑车",
"黑卒","黑炮","红车","红马","红相","红仕","红帅","红仕","红相","红马","红车",
         "红兵","红炮"]
```

```
imgs=[PhotoImage(file='bmp\\'+chessname[i]+'.png')for i in range(0,22)]
chessmap=[[-1,-1,-1,-1,-1,-1,-1,-1,-1,-1]for y in range(10)]
dict_ChessName={}          #定义一个字典
LocalPlayer="红"           #LocalPlayer 记录自己是红方还是黑方
first=True                 #区分是第 1 次还是第 2 次选中的棋子
IsMyTurn=True
rect1=0
rect2=0
firstChessid=0
```

程序运行时，首先调用 DrawBoard()和 LoadChess()加载棋盘图片和棋子到 Canvas 中；LoadChess()初始化游戏区中各个棋子的位置，红方在南，黑方在北，并且在 chessmap 列表中按坐标记录每个棋子图像的 id；然后绑定 Canvas 鼠标事件函数 callback()，也就是鼠标单击游戏画面时的处理函数，在此函数中处理游戏的走棋吃子过程。

```
img1=PhotoImage(file='bmp\\棋盘.png')
def DrawBoard():                                        #画棋盘
    p1=cv.create image((0,0),image=img1)
    cv.coords(p1,(360,400))
def LoadChess():                                        #加载棋子
    global chessmap
    #黑方 16 个棋子
    for i in range(0,9):
            #"黑车","黑马","黑象","黑仕","黑将","黑仕","黑象","黑马","黑车"
      img=imgs[i]
      id=cv.create_image((60+76*i,54),image=img)        #76×76 棋盘格子大小
      dict_ChessName[id]=chessname[i];                  #图像对应的是哪种棋子
      chessmap[i][0]=id                                 #图像 id
    for i in range(0,5):                                #5 个卒
      img=imgs[9]                                       #卒图像
      id=cv.create_image((60+76*2*i,54+3*76),image=img)
                                                        #76×76 棋盘格子大小
      chessmap[i*2][3]=id
      dict ChessName[id]="黑卒";                         #图像对应的是哪种棋子
    img=imgs[10]                                        #黑方炮
    id=cv.create_image((60+76*1,54+2*76),image=img) #76×76 棋盘格子大小
    chessmap[1][2]=id
    dict_ChessName[id]="黑炮";                      #图像对应的是哪种棋子
    id=cv.create_image((60+76*7,54+2*76),image=img) #76×76 棋盘格子大小
    chessmap[7][2]=id
    dict_ChessName[id]="黑炮";  #图像对应的是哪种棋子
    #红方 16 个棋子
    for i in range(0,9):
            #"红车","红马","红相","红仕","红帅","红仕","红相","红马","红车"
```

```
        img=imgs[i+11]
        id=cv.create_image((60+76*i,54+9*76),image=img)  #76×76 棋盘格子大小
        dict_ChessName[id]=chessname[i+11];      #图像对应的是哪种棋子
        chessmap[i][9]=id                        #图像 id
    for i in range(0,5):                         #5 个兵
        img=imgs[20]                             #兵图像
        id=cv.create_image((60+76*2*i,54+6*76),image=img)  #76×76 棋盘格子大小
        chessmap[i*2][6]=id                      #图像 id
        dict_ChessName[id]=chessname[20];        #图像对应的是哪种棋子
    img=imgs[21]                                 #红方炮
    id=cv.create_image((60+76*1,54+7*76),image=img) #76×76 棋盘格子大小
    chessmap[1][7]=id
    dict_ChessName[id]="红炮";                   #图像对应的是哪种棋子
    id=cv.create_image((60+76*7,54+7*76),image=img) #76×76 棋盘格子大小
    chessmap[7][7]=id
    dict_ChessName[id]="红炮";                   #图像对应的是哪种棋子
#———————————————————————————————————————————————
DrawBoard()                                      #画棋盘
LoadChess()                                      #加载棋子
#———————————————————————————————————————————————
print(dict_ChessName)
cv.bind("<Button-1>", callback)
cv.pack()
lable1=Label(root, fg='red', bg='white', text="红方先走")   #提示信息标签
lable1['text']="红方先走 1"
lable1.pack()
root.mainloop()
```

游戏区的单击事件处理用户走棋过程。在用户走棋时，首先要选中自己的棋子（第 1 次选择棋子），所以有必要判断是否单击成对方棋子了。如果是自己的棋子，则 firstChessid 记录用户选择的棋子，同时棋子被加上红色框线示意被选中。

当用户选过己方棋子后，单击对方棋子（secondChessid 记录用户第 2 次选择的棋子，被加上黄色框线），则是吃子，如果将或帅被吃掉，则游戏结束。当然，第 2 次选择棋子有可能是用户改变主意，选择自己的另一棋子，则 firstChessid 重新记录用户选择的己方棋子。

当用户选过己方棋子后，在单击的位置无棋子，则处理没有吃子的走棋过程。调用 IsAbleToPut(CurSelect, x, y)判断是否能走棋，如果符合走棋规则，移动棋子，修改 chessmap 记录的棋子信息。

```
def callback(event):         #走棋 picBoard_MouseClick
    global LocalPlayer
    global chessmap
    global rect1,rect2       #选中框图像 id
    global firstChessid,secondChessid
    global x1,x2,y1,y2
    global first
    print("clicked at", event.x, event.y, LocalPlayer)
```

```
x=(event.x-14)//76                  #换算棋盘坐标
y=(event.y-14)//76
print("clicked at", x, y, LocalPlayer)

if (first):                         #第 1 次单击棋子
    x1=x;
    y1=y;
    firstChessid=chessmap[x1][y1]
    if  not(chessmap[x1][y1]==-1):  #此位置不空，有棋子
        player=dict ChessName[firstChessid][0]
                                    #获取单击棋子的颜色，例如“红马”取红
        if(player!=LocalPlayer):    #颜色不同
            print("单击成对方棋子了!");
            return
        print("第 1 次单击",firstChessid)
        first=False;
        rect1=cv.create_rectangle(60+76*x-40, 54+y*76-38,
        60+76*x+80-40,54+y*76+80-38,outline="red")   #画选中标记框
else:                               #第 2 次单击
    x2=x;
    y2=y;
    secondChessid=chessmap[x2][y2]
    #目标处如果是自己的棋子，则换上次选择的棋子
    if not(chessmap[x2][y2]==-1):   #此位置不空，有棋子
        player=dict_ChessName[secondChessid][0] #获取单击棋子的颜色
        if(player==LocalPlayer):    #如果是自己的棋子，则换上次选择的棋子
            firstChessid=chessmap[x2][y2]
            print("第 2 次单击",firstChessid)
            cv.delete(rect1);       #取消上次选择的棋子标记框
            x1=x;
            y1=y;
            #设置选择的棋子颜色
            rect1=cv.create_rectangle(60+76*x-40,54+y*76-38,
            60+76*x+80-40,54+y*76+80-38,outline="red")    #画选中标记框
            print("第 2 次单击",firstChessid)
            return;
        else:                       #在落子目标处画框
            rect2=cv.create_rectangle(60+76*x-40,54+y*76-38,
            60+76*x+80-40,54+y*76+80-38,outline="yellow") #目标处画框
    #目标处没棋子，移动棋子
    print("kkkkk",firstChessid)
    if(chessmap[x2][y2]==" " or chessmap[x2][y2]==-1):
                                               #目标处没棋子，移动棋子
```

```
        print("目标处没棋子，移动棋子",firstChessid,x2,y2,x1,y1)
        if(IsAbleToPut(firstChessid,x2,y2,x1,y1)): #判断是否可以走棋
            print("can 移动棋子",x1,y1)
            cv.move(firstChessid,76*(x2-x1),76*(y2-y1));
            #*****************************************************
            #在 map 中取掉原棋子
            chessmap[x1][y1]=-1;
            chessmap[x2][y2]=firstChessid
            cv.delete(rect1);                 #删除选中标记框
            cv.delete(rect2);                 #删除目标标记框
            #*****************************************************
            first=True;
            SetMyTurn(False);                 #该对方了
        else:
            #错误走棋
            print("不符合走棋规则");
            showinfo(title="提示",message="不符合走棋规则")
        return;
    else:
    #目标处有棋子，可以吃子
        if(not(chessmap[x2][y2]==-1) and IsAbleToPut(firstChessid,x2,
        y2,x1,y1)):                           #可以吃子
            first=True;
            print("can 吃子",x1,y1)
            cv.move(firstChessid,76*(x2-x1),76*(y2-y1));
            #*****************************************************
            #在 map 中取掉原棋子
            chessmap[x1][y1]=-1;
            chessmap[x2][y2]=firstChessid
            cv.delete(secondChessid);
            cv.delete(rect1);
            cv.delete(rect2);
            #*****************************************************
            if(dict ChessName[secondChessid][1]=="将"):  #"将"
                showinfo(title="提示",message="红方你赢了")
                return;
            if(dict_ChessName[secondChessid][1]=="帅"):  #"帅"
                showinfo(title="提示",message="黑方你赢了")
                return;
            #send
            SetMyTurn(False);                        #该对方了
        else:                                        #不能吃子
          print("不能吃子");
          lable1['text']="不能吃子"
```

```
            cv.delete(rect2);                                        #删除目标标记框
```

SetMyTurn()设置该哪方走棋，LocalPlayer 记录轮到哪方走棋，并在标签上显示提示信息。

```
def SetMyTurn(flag):
  global LocalPlayer
  IsMyTurn=flag
  if LocalPlayer=="红" :
     LocalPlayer="黑"
     lable1['text']="轮到黑方走"
  else:
     LocalPlayer="红"
     lable1['text']="轮到红方走"
```

def IsAbleToPut(id,x,y,oldx,oldy)实现判断是否能走棋并返回逻辑值，其代码较复杂。

```
def IsAbleToPut(id, x, y, oldx, oldy):
   #oldx, oldy 棋子在棋盘的原坐标
   #x, y       棋子移动到棋盘的新坐标
   qi name=dict ChessName[id][1]
             #取字符串中的第 2 个字符，例如“黑将”中的“将”，从而得到棋子类型
   #“将”或“帅”的走棋判断
   if(qi name=="将" or qi name=="帅"):
      if((x-oldx)*(y-oldy)!=0):
         return False;
      if(abs(x-oldx)>1 or abs(y-oldy)>1):
         return False;
      if(x<3 or x>5 or (y>=3 and y<=6)):
         return False;
      return True;
   #“仕”或“士”的走棋判断
   if(qi_name=="士" or qi_name=="仕"):
      if((x-oldx)*(y-oldy)==0):
         return False;
      if(abs(x-oldx)>1 or abs(y-oldy)>1):
         return False;
      if(x<3 or x>5 or (y>=3 and y<=6)):
         return False;
      return True;
   #“象”或“相”的走棋判断
   if(qi name=="象" or qi name=="相"):
      if((x-oldx)*(y-oldy)==0):
         return False;
      if(abs(x-oldx)!=2 or abs(y-oldy)!=2):
         return False;
```

```
        if(y<5 and qi_name=="相"):              #过河
            return False;
        if(y>=5 and qi name=="象"):             #过河
            return False;
        i=0; j=0;                               #i、j 必须有初始值
        if(x-oldx==2):
            i=x-1;
        if(x-oldx==-2):
            i=x+1;
        if(y-oldy==2):
            j=y-1;
        if(y-oldy==-2):
            j=y+1;
        if(chessmap[i][j]!=-1):                 #蹩象腿
           return False;
        return True;
    #"马"的走棋判断
    if(qi_name=="马"):
        if(abs(x-oldx)*abs(y-oldy)!=2):
            return False;
        if(x-oldx==2):
            if(chessmap[x-1][oldy]!=-1):        #蹩马腿
                return False;
        if(x-oldx==-2):
            if(chessmap[x+1][oldy]!=-1):        #蹩马腿
                return False;
        if(y-oldy==2):
            if(chessmap[oldx][y-1]!=-1):        #蹩马腿
                return False;
        if(y-oldy==-2):
            if(chessmap[oldx][y+1]!=-1):        #蹩马腿
                return False;
        return True;
    #“车”的走棋判断
    if(qi_name=="车"):
        #判断是否为直线
        if((x-oldx)*(y-oldy)!=0):
           return False;
        #判断是否隔有棋子
        if(x!=oldx):
            if(oldx>x):
                t=x;
                x=oldx;
                oldx=t;
```

```
            for i in range(oldx,x+1):
                if(i!=x and i!=oldx):
                    if(chessmap[i][y]!=-1):
                        return False;
        if(y!=oldy):
            if(oldy>y):
                t=y;
                y=oldy;
                oldy=t;
            for  j in range(oldy,y+1):
                if(j!=y and j!=oldy):
                    if (chessmap[x][j]!=-1):
                        return False;
        return True;
    #“炮”的走棋判断
    if(qi name=="炮"):
        swapflagx=False;
        swapflagy=False;
        if((x-oldx)*(y-oldy)!=0):
            return False;
        c=0;
        if(x!=oldx):
            if(oldx>x):
               t=x;
               x=oldx;
               oldx=t;
               swapflagx=True;
            for  i in range(oldx,x+1): #for (i=oldx; i<=x; i+=1):
                if(i!=x and i!=oldx):
                    if(chessmap[i][y]!=-1):
                        c=c+1;
        if(y!=oldy):
            if(oldy>y):
               t=y;
               y=oldy;
               oldy=t;
               swapflagy=True;
            for  j in range(oldy,y+1): #for(j=oldy; j<=y; j+=1):
                if(j!=y and j!=oldy):
                    if(chessmap[x][j]!=-1):
                        c=c+1;
        if(c>1):
           return False;           #与目标处间隔 1 个以上的棋子
```

```
        if(c==0):                              #与目标处无间隔棋子
            if(swapflagx==True):
                t=x;
                x=oldx;
                oldx=t;
            if(swapflagy==True):
                t=y;
                y=oldy;
                oldy=t;
            if(chessmap[x][y]!=-1):
                return False;
        if(c==1):                              #与目标处间隔 1 个棋子
            if(swapflagx==True):
                t=x;
                x=oldx;
                oldx=t;
            if(swapflagy==True):
                t=y;
                y=oldy;
                oldy=t;
            if(chessmap[x][y]==-1):            #如果目标处无棋子，则不能走此步
                return False;
        return True;
    #“卒”或“兵”的走棋判断
    if(qi name == "卒" or qi name=="兵"):
        if((x-oldx)*(y-oldy)!=0):              #不是直线走棋
            return False;
        if(abs(x-oldx)>1 or abs(y-oldy)>1):
                                               #走多步，不符合兵只能走一步
            return False;
        if(y>=5 and (x-oldx)!=0 and qi_name=="兵"):
                                               #未过河且横向走棋
            return False;
        if(y<5 and (x-oldx)!=0 and qi_name=="卒"):
                                               #未过河且横向走棋
            return False;
        if(y-oldy>0 and qi_name=="兵"):        #后退
            return False;
        if(y-oldy<0 and qi name=="卒"):        #后退
            return False;
        return True;
    return True;
```

其运行效果如图 15-5 和图 15-6 所示。这个游戏，双方在本机轮下，读者可以根据网

络五子棋的 UDP 通信知识完善本游戏，从而实现网络版中国象棋。

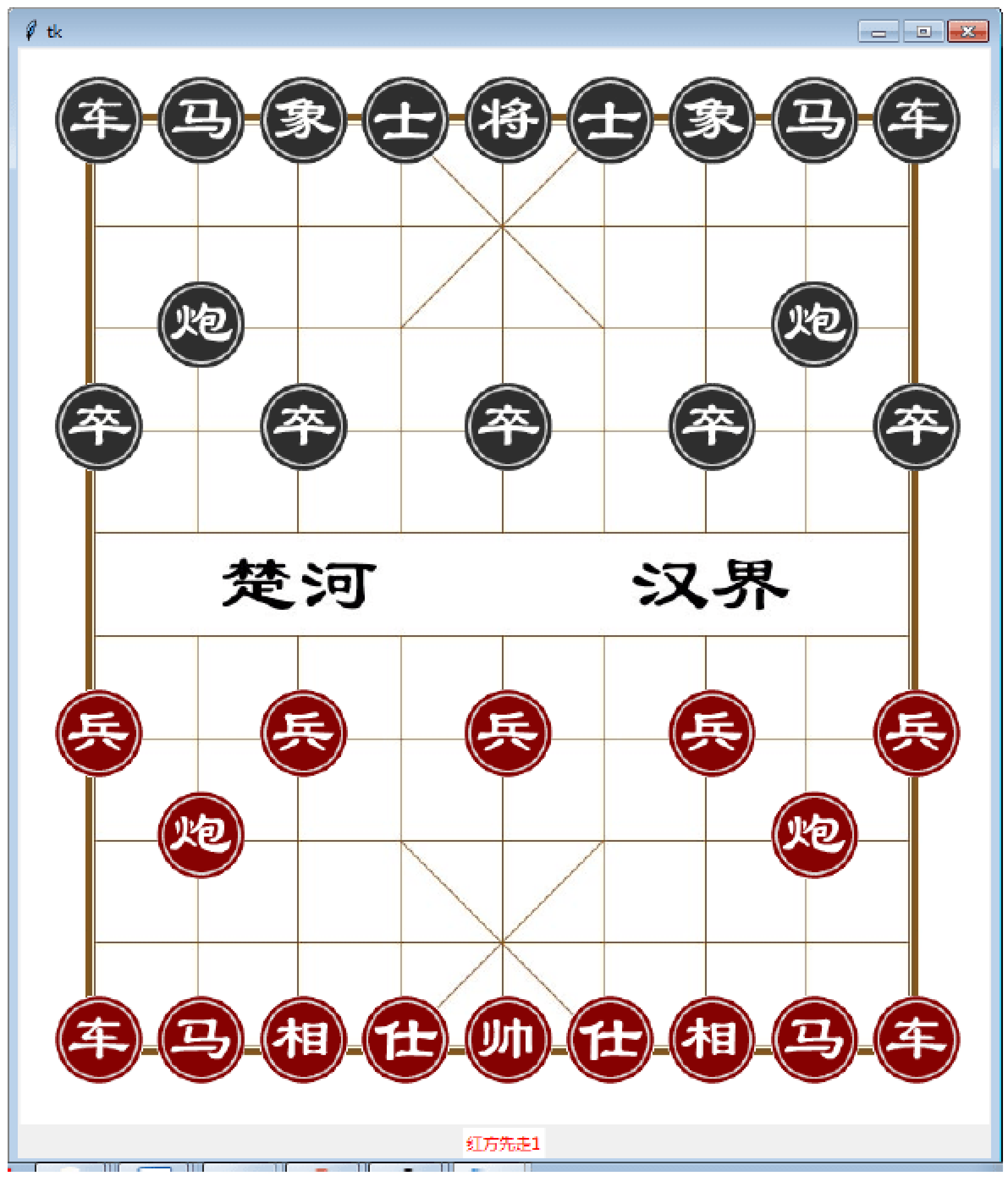


图 15-5　中国象棋初始界面

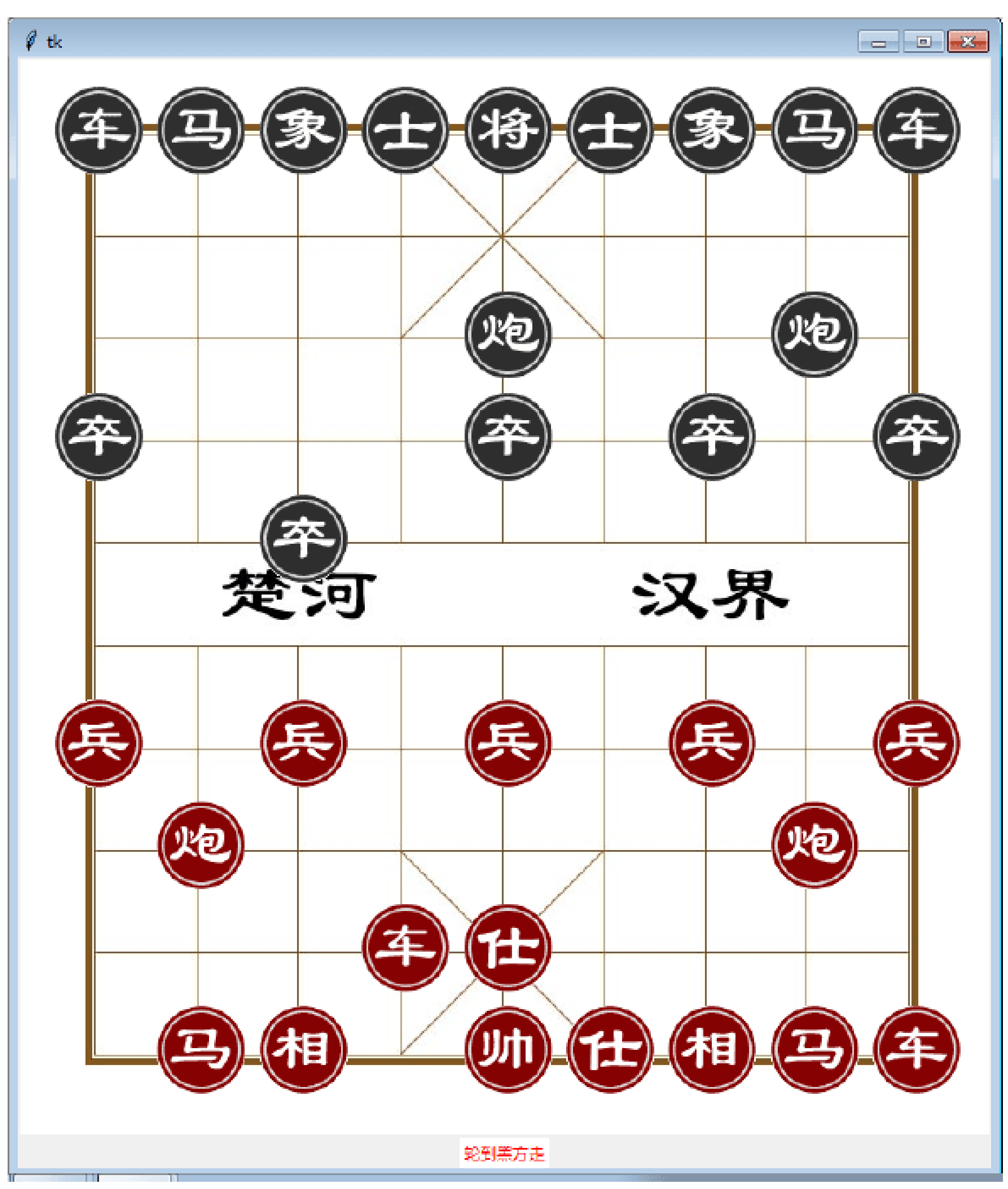


图 15-6　中国象棋运行界面

第16章 娱乐游戏——人物拼图游戏

16.1　人物拼图游戏介绍

视频讲解

拼图游戏将一幅图片分割成若干拼块，并将它们随机打乱顺序，当将所有拼块都放回原位置时就完成了拼图（游戏结束）。

本人物拼图游戏为 3 行 3 列，拼块以随机顺序排列，玩家通过用鼠标单击空白块四周来交换它们的位置，直到所有拼块都回到原位置。拼图游戏的运行界面如图 16-1 所示。

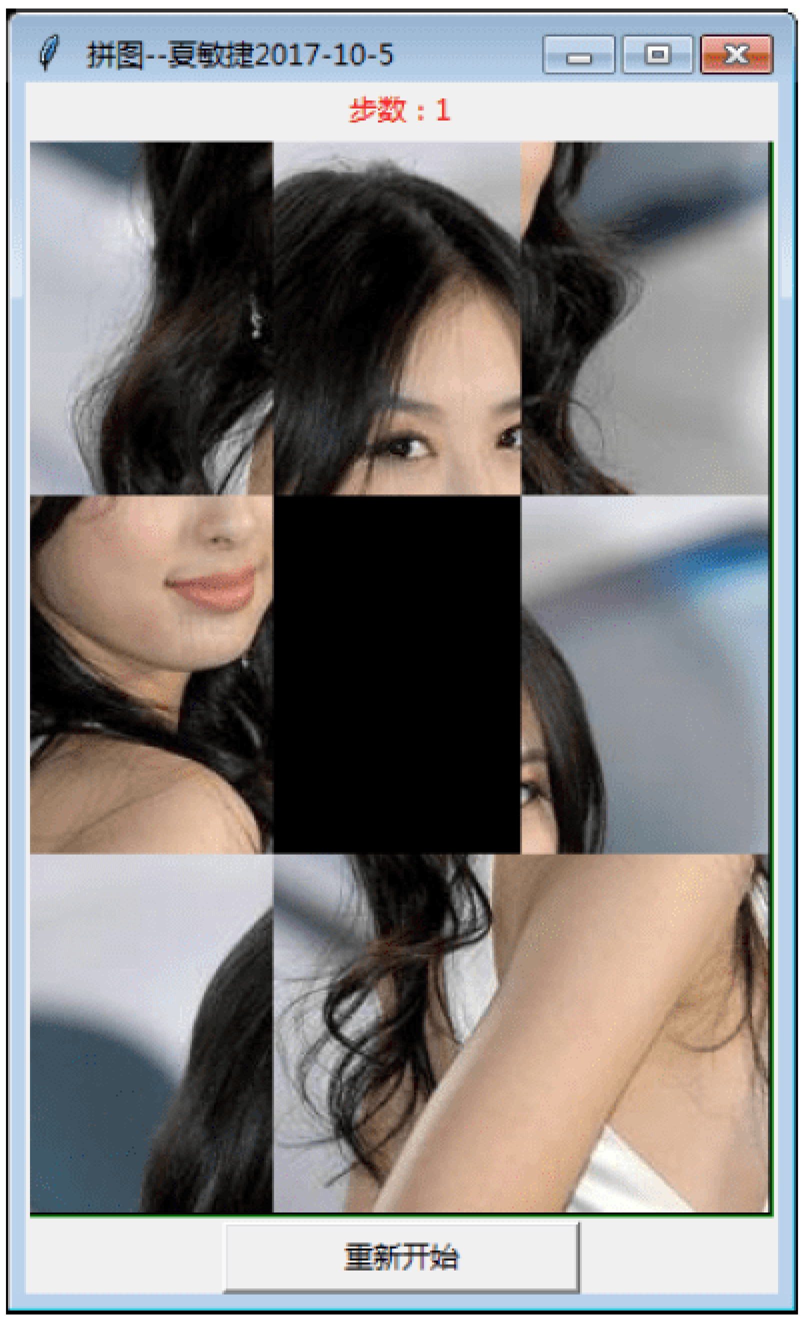


图 16-1　拼图游戏的运行界面

16.2　程序设计的思路

游戏程序首先将图片分割成相应的 3 行 3 列的拼块，并按顺序编号，动态生成一个大小为 3×3 的列表 board，存放 0～8 的数，每个数字代表一个拼块（例如 3×3 的游戏拼块的编号如图 16-2 所示），其中 8 号拼块不显示。



图 16-2　拼块编号示意图

游戏开始时，随机打乱数组 board，例如 board[0][0]是 5 号拼块，则在左上角显示编号是 5 的拼块。根据玩家用鼠标单击的拼块和空白块所在位置来交换 board 数组对应的元素，最后通过判断元素的排列顺序来判断是否已经完成游戏。

16.3　关键技术

16.3.1　复制和粘贴图像区域

使用 crop()方法可以从一幅图像中裁剪指定区域，例如：

```
from PIL import Image
im=Image.open("D:\\test.jpg")
box=(100,100,400,400)
region=im.crop(box)
```

该区域使用四元组来指定。四元组的坐标是(左,上,右,下)。在 PIL 中指定坐标系的左上角坐标为(0,0)。用户可以旋转上面代码获取的区域，然后使用 paste()方法将该区域放回去，具体实现如下：

```
region=region.transpose(Image.ROTATE_180)  #逆时针旋转 180°
im.paste(region,box)
```

16.3.2　调整尺寸和旋转

如果要调整一幅图像的尺寸，可以调用 resize()方法。该方法的参数是一个元组，用来指定新图像的大小：

```
out=im.resize((128,128))
```

如果要旋转一幅图像，可以使用逆时针方式表示旋转角度，然后调用 rotate()方法：

```
out=im.rotate(45)                          #逆时针旋转 45°
```

16.3.3　转换成灰度图像

对于彩色图像，不管其图像格式是 PNG、BMP 还是 JPG，在 PIL 中使用 Image 模块的 open()函数打开后，返回的图像对象的模式都是“RGB”。对于灰度图像，不管其图像格式是 PNG、BMP 还是 JPG，打开后模式都为“L”。

对于 PNG、BMP 和 JPG 彩色图像格式之间的互相转换，都可以通过 Image 模块的 open()和 save()函数来完成。具体来说就是，在打开这些图像时 PIL 会将它们解码为三通道的“RGB”图像，用户可以基于这个“RGB”图像对其进行处理，处理完毕后使用 save()函数可以将处理结果保存成 PNG、BMP 和 JPG 中的任何格式，这样也就完成了几种格式之间的转换。当然，对于不同格式的灰度图像，也可以通过类似途径完成，只是 PIL 解码后是模式为“L”的图像。

这里详细介绍一下 Image 模块的 convert()函数，用于不同模式图像之间的转换。

convert()函数有 3 种形式的定义，它们的定义形式如下：

```
im.convert(mode)
im.convert('P', **options)
im.convert(mode, matrix)
```

使用不同的参数，将当前的图像转换为新的模式（在 PIL 中有 9 种不同模式，分别为 1、L、P、RGB、RGBA、CMYK、YCbCr、I、F），并产生新的图像作为返回值。

例如：

```
from PIL import Image          #或直接 import Image
im=Image.open('a.jpg')
im1=im.convert('L')            #将图片转换成灰度图
```

“L”模式的图像为灰色图像，它的每个像素用 8 个 bit 表示，0 表示黑，255 表示白，其他数字表示不同的灰度。在 PIL 中，从“RGB”模式转换为“L”模式是按照下面的公式进行的：

$$L=R*299/1000+G*587/1000+ B*114/1000$$

打开图片并转换成灰度图的方法如下：

```
im=Image.open('a.jpg').convert('L')
```

如果转换成黑白图像（二值图像），也就是模式为“1”的图像（非黑即白），它的每个像素用 8 个 bit 表示，0 表示黑，255 表示白。下面将彩色图像转换为黑白图像。

```
from PIL import Image          #或直接 import Image
im=Image.open('a.jpg')
im1=im.convert('1')            #将彩色图像转换成黑白图像
```

16.3.4 对像素进行操作

getpixel(x,y)用于获取指定像素的颜色，如果图像为多通道，则返回一个元组。该方法的执行比较慢，如果用户需要使用 Python 处理图像中的较大部分数据，可以使用像素访问对象（load()或者 getdata()方法）。putpixel(xy,color)可以改变单个像素点的颜色。

```
img=Image.open("smallimg.png")
img.getpixel((4,4))                #获取像素的颜色
img.putpixel((4,4),(255,0,0))      #改变单个像素点为红色
img.save("img1.png","png")
```

说明：getpixel()得到图片 img 的坐标为(4,4)的像素点，putpixel()将坐标为(4,4)的像素点变为(255,0,0)的颜色，即红色。

16.4 程序设计的步骤

16.4.1 Python 处理图片切割

使用 PIL 库的 Image 模块中的 crop()方法可以从一幅图像中裁剪指定区域，该区域使用四元组来指定，四元组的坐标是(左,上,右,下)。在 PIL 中指定坐标系的左上角坐标为(0,0)。其具体实现如下：

```
from PIL import Image
img=Image.open(r'C:\woman.jpg')
box=(100,100,400,400)
region=img.crop(box)               #裁切图片
#保存裁切后的图片
region.save('crop.jpg')
```

在本游戏中需要把图片分割成 3 行×3 列的图片块，在上面的基础上指定不同的区域即可裁剪、保存。为了更通用一些，编成 splitimage(src,rownum,colnum,dstpath)函数，实现将指定的 src 图片文件分割成 rownum×colnum 数量的小图片块。其具体实现如下：

```
import os
from PIL import Image
def splitimage(src, rownum, colnum, dstpath):
    img=Image.open(src)
    w, h=img.size            #图片大小
    if rownum<=h and colnum<=w:
        print('Original image info: %sx%s, %s, %s' % (w, h, img.format,
        img.mode))
```

```
        print('开始处理图片切割, 请稍候…')
        s=os.path.split(src)
        if dstpath=='':              #没有输入路径
          dstpath=s[0]               #使用源图片所在目录 s[0]
        fn=s[1].split('.')           #s[1]是源图片文件名
        basename=fn[0]               #主文件名
        ext=fn[-1]                   #扩展名
        num=0
        rowheight = h                #rownum
        colwidth = w                 #colnum
        for r in range(rownum):
            for c in range(colnum):
                box=(c*colwidth, r*rowheight, (c+1)*colwidth,(r+1)*rowheight)
                img.crop(box).save(os.path.join(dstpath, basename+' ' +
                str(num)+'.'+ext))
                num=num+1
        print('图片切割完毕，共生成 %s 张小图片。' % num)
    else:
        print('不合法的行列切割参数！')
src=input('请输入图片文件路径：')
#src="C:\woman.png"
if os.path.isfile(src):
    dstpath=input('请输入图片输出目录（不输入路径则表示使用源图片所在目录）：')
    if (dstpath=='') or os.path.exists(dstpath):
        row=int(input('请输入切割行数：'))
        col=int(input('请输入切割列数：'))
        if row>0 and col>0:
            splitimage(src, row, col, dstpath)
        else:
            print('无效的行列切割参数！')
    else:
        print('图片输出目录 %s 不存在！' % dstpath)
else:
    print('图片文件 %s 不存在！' % src)
```

运行结果如下：

```
请输入图片文件路径：C:\woman.png
请输入图片输出目录（不输入路径则表示使用源图片所在目录）：
请输入切割行数：3
请输入切割列数：3
Original image info: 283x212, PNG, RGBA
开始处理图片切割, 请稍候…
图片切割完毕，共生成 9 张小图片。
```

16.4.2 游戏的逻辑实现

视频讲解

❶ 定义常量及加载图片

```
from tkinter import *
from tkinter.messagebox import *
import random
#定义常量
#画布的尺寸
WIDTH=312
HEIGHT=450
#图像块的边长
IMAGE WIDTH=WIDTH // 3
IMAGE HEIGHT=HEIGHT // 3
#游戏的行/列数
ROWS=3
COLS=3
#移动步数
steps=0
#保存所有图像块的列表
board=[[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]]
root=Tk('拼图 2017')
root.title("拼图--夏敏捷 2017-10-5")
#载入外部事先生成的 9 个小图像块
Pics=[]
for i in range(9):
   filename="woman "+str(i)+".png"
   Pics.append(PhotoImage(file=filename))
```

❷ 图像块类

每个图像块(拼块)是一个 Square 对象，具有 draw 功能，即将本拼块图片绘制到 Canvas 上。orderID 属性是每个图像块（拼块）对应的编号。

```
#图像块（拼块）类
class Square:
    def   init   (self, orderID):
        self.orderID=orderID
    def draw(self, canvas, board pos):
        img=Pics[self.orderID]
        canvas.create_image(board_pos, image=img)
```

❸ 初始化游戏

random.shuffle(board)打乱二维列表只能按行进行，所以使用一维列表来实现编号的打

乱。在打乱图像块后，根据编号生成对应的图像块（拼块）到 board 列表中。

```
def init board():
    #打乱图像块
    L=list(range(9))                    #L 列表中[0,1,2,3,4,5,6,7,8]
    random.shuffle(L)
    #填充拼图板
    for i in range(ROWS):
      for j in range(COLS):
           idx=i*ROWS+j
           orderID=L[idx]
           if orderID is 8:             #8 号拼块不显示，所以存为 None
               board[i][j]=None
           else:
               board[i][j]=Square(orderID)
```

❹ **绘制游戏界面中的各元素**

```
def drawBoard(canvas):
    #画黑框
    canvas.create polygon((0, 0, WIDTH, 0, WIDTH, HEIGHT, 0, HEIGHT),
    width=1,outline='Black')
    #画所有图像块
    for i in range(ROWS):
        for j in range(COLS):
            if board[i][j] is not None:
                board[i][j].draw(canvas, (IMAGE_WIDTH*(j+0.5),IMAGE_
                HEIGHT*(i+0.5)))
```

❺ **鼠标事件**

将单击位置换算成拼图板上的棋盘坐标，如果单击空位置，什么也不移动，否则依次检查被单击当前图像块的上、下、左、右是否有空位置，如果有，就移动当前图像块。

```
def mouseclick(pos):
    global steps
    #将单击位置换算成拼图板上的棋盘坐标
    r=int(pos.y // IMAGE HEIGHT)
    c=int(pos.x // IMAGE WIDTH)
    if r < 3 and c < 3:          #单击位置在拼图板内才移动图片
        if board[r][c] is None: #单击空位置，什么也不移动
            return
        else:
        #依次检查被单击当前图像块的上、下、左、右是否有空位置，如果有，就移动当前图像块
            current_square=board[r][c]
            if r-1>=0 and board[r-1][c] is None:     #判断上面
```

```
                board[r][c]=None
                board[r-1][c]=current_square
                steps+=1
            elif c+1<=2 and board[r][c+1] is None:   #判断右面
                board[r][c]=None
                board[r][c+1]=current square
                steps+=1
            elif r+1<=2 and board[r+1][c] is None:   #判断下面
                board[r][c]=None
                board[r+1][c]=current square
                steps+=1
            elif c-1>=0 and board[r][c-1] is None:   #判断左面
                board[r][c]=None
                board[r][c-1]=current square
                steps+=1
            #print(board)
            label1["text"]="步数: "+str(steps)
            cv.delete('all')                         #清除画布上的内容
            drawBoard(cv)
    if win():
        showinfo(title="恭喜",message="你成功了！")
```

❻ **输赢的判断**

判断拼块的编号是否为有序的，如果不是有序的，则返回 False。

```
def win():
   for i in range(ROWS):
        for j in range(COLS):
          if board[i][j] is not None  and  board[i][j].orderID!=i * ROWS + j:
              return False
   return True
```

❼ **重置游戏**

```
def play game():
    global steps
    steps=0
    init_board()
```

❽ **“重新开始”按钮的单击事件**

```
def callBack2():
   print("重新开始")
   play_game()
```

```
    cv.delete('all')        #清除画布上的内容
    drawBoard(cv)
```

❾ 主程序

```
#设置窗口
cv=Canvas(root, bg='green', width=WIDTH, height=HEIGHT)
b1=Button(root,text="重新开始",command=callBack2,width=20)
label1=Label(root,text="步数: "+str(steps),fg="red",width=20)
label1.pack()
cv.bind("<Button-1>", mouseclick)
cv.pack()
b1.pack()
play game()
drawBoard(cv)
root.mainloop()
```

至此完成人物拼图游戏设计。

第 17 章 基于 Pygame 的游戏设计

Pygame 最初由 Pete Shinners 开发，它是一个跨平台的 Python 模块，专为电子游戏设计，包含图像、声音功能和网络支持，这些功能使开发者很容易用 Python 写一个游戏。虽然不使用 Pygame 也可以写一个游戏，但如果能充分利用 Pygame 库中已经写好的代码，开发要容易得多。Pygame 能把游戏设计者从低级语言（例如 C 语言）的束缚中解放出来，专注于游戏逻辑本身。

由于 Pygame 很容易使用且跨平台，所以在游戏开发中十分受欢迎。因为 Pygame 是开放源代码的软件，也促使一大批游戏开发者为完善和增强它的功能而努力。

17.1 Pygame 基础知识

17.1.1 安装 Pygame 库

视频讲解

在开发 Pygame 程序之前，需要安装 Pygame 库。用户可以通过 Pygame 的官方网站“http://www.pygame.org/download.shtml”下载源文件，安装指导也可以在相应页面找到。

一旦安装了 Pygame，就可以在 IDLE 交互模式中输入以下语句检验是否安装成功：

```
>>> import pygame
>>> print(pygame.ver)
1.9.2a0
```

1.9.2 是 Pygame 的最新版本，读者也可以找一找其他更新的版本。

17.1.2 Pygame 的模块

Pygame 有大量可以被独立使用的模块。对于计算机的常用设备，都有对应的模块进行

控制，另外还有其他一些模块，例如 pygame.display 是显示模块、pygame.keyboard 是键盘模块、pygame.mouse 是鼠标模块，如表 17-1 所示。

表 17-1　Pygame 软件包中的模块

模　块　名	功　　能
pygame.cdrom	访问光驱
pygame.cursors	加载光标
pygame.display	访问显示设备
pygame.draw	绘制形状、线和点
pygame.event	管理事件
pygame.font	使用字体
pygame.image	加载和存储图片
pygame.joystick	使用游戏手柄或者类似的东西
pygame.key	读取键盘按键
pygame.mixer	声音
pygame.mouse	鼠标
pygame.movie	播放视频
pygame.music	播放音频
pygame.overlay	访问高级视频叠加
pygame	Python 模块，专为电子游戏设计
pygame.rect	管理矩形区域
pygame.sndarray	操作声音数据
pygame.sprite	操作移动图像
pygame.surface	管理图像和屏幕
pygame.surfarray	管理点阵图像数据
pygame.time	管理时间和帧信息
pygame.transform	缩放和移动图像

建立 Pygame 项目和建立其他 Python 项目的方法一样，在 IDLE 或文本编辑器中新建一个空文档，需要告诉 Python 该程序用到了 Pygame 模块。

为了实现此目的，这里使用一个 import 指令，该指令告诉 Python 载入外部模块。例如输入下面两行在新项目中引入必要的模块：

```
import pygame, sys, time, random
from pygame.locals import *
```

第 1 行引入 Pygame 的主要模块、sys 模块、time 模块和 random 模块。

第 2 行告诉 Python 载入 pygame.locals 的所有指令使它们成为原生指令，这样在使用这些指令时就不需要用全名调用。

由于硬件和游戏的兼容性或者请求的驱动没有安装的问题，有些模块可能在某些平台上不存在，可以用 None 测试一下。例如测试字体是否载入：

```
if pygame.font is None:
```

```
    print("The font module is not available!")
    pygame.quit()                          #如果没有则退出 Pygame 的应用环境
```

下面对常用模块进行简要说明。

❶ **pygame.surface**

该模块中有一个 surface()函数，surface()函数的一般格式如下：

```
pygame.surface((width, height), flags=0, depth=0, masks=none)
```

它返回一个新的 surface 对象。这里的 surface 对象是一个有确定尺寸的空图像，可以用它进行图像的绘制与移动。

❷ **pygame.locals**

在 pygame.locals 模块中定义了 Pygame 环境中用到的各种常量，而且包括事件类型、按键和视频模式等的名字，在导入所有内容（from pygame.locals import *）时用起来很安全。

如果用户知道需要的内容，也可以导入具体的内容（例如 from pygame.locals import FULLSCREEN）。

❸ **pygame.display**

pygame.display 模块包括处理 Pygame 显示方式的函数，其中包括普通窗口和全屏模式。

游戏程序通常需要下面的函数：

1）flip()/update()

（1）flip()：更新显示。一般来说，在修改当前屏幕的时候要经过两步，首先需要对 get_surface()函数返回的 surface 对象进行修改，然后调用 pygame.display.flip()更新显示以反映所做的修改。

（2）update()：在只想更新屏幕一部分的时候使用 update()函数，而不是 flip()函数。

2）set_mode()

该函数建立游戏窗口，返回 surface 对象。它有 3 个参数，第 1 个参数是元组，用于指定窗口的尺寸；第 2 个参数是标志位，具体含义如表 17-2 所示，例如 FULLSCREEN 表示全屏，默认值为不对窗口进行设置，读者可根据需要选用；第 3 个参数为色深，用于指定窗口的色彩位数。

表 17-2　set_mode 的窗口标志位的参数取值

窗口标志位	功　能
FULLSCREEN	创建一个全屏窗口
DOUBLEBUF	创建一个“双缓冲”窗口，建议在 HWSURFACE 或者 OPENGL 时使用
HWSURFACE	创建一个硬件加速的窗口，必须和 FULLSCREEN 同时使用
OPENGL	创建一个 OPENGL 渲染的窗口
RESIZABLE	创建一个可以改变大小的窗口
NOFRAME	创建一个没有边框的窗口

3）set_caption()

该函数设定游戏程序的标题。当游戏以窗口模式（对应于全屏）运行时尤其有用，因

为该标题会作为窗口的标题。

4）get_surface()

该函数返回一个可用来画图的 surface 对象。

❹ pygame.font

pygame.font 模块用于表现不同字体，可以用于文本。

❺ pygame.sprite

pygame.sprite 模块有两个非常重要的类——sprite 精灵类和 group 精灵组。

sprite 精灵类是所有可视游戏的基类。为了实现自己的游戏对象，需要子类化 sprite，覆盖它的构造函数，以设定 image 和 rect 属性（决定 sprite 的外观和放置的位置)，再覆盖 update()方法。在 sprite 需要更新的时候可以调用 update()方法。

group 精灵组的实例用作 sprite 精灵对象的容器。在一些简单的游戏中，只要创建名为 sprites、allsprite 或是其他类似的组，然后将所有 sprite 精灵对象添加到上面即可。当 group 精灵组对象的 update()方法被调用时会自动调用所有 sprite 精灵对象的 update()方法。group 精灵组对象的 clear()方法用于清理它包含的所有 sprite 对象（使用回调函数实现清理)，group 精灵组对象的 draw()方法用于绘制所有的 sprite 对象。

❻ pygame.mouse

该模块用来管理鼠标。

- pygame.mouse.set_visible(False/true)：隐藏/显示鼠标光标。
- pygame.mouse.get_pos()：获取鼠标位置。

❼ pygame.event

pygame.event 模块会追踪鼠标单击、鼠标移动、按键按下和释放等事件。其中，pygame .event.get ()可以获取最近事件列表。

❽ pygame.image

这个模块用于处理保存在 GIF、PNG 或者 JPEG 内的图形，用户可以用 load()函数来读取图像文件。

17.2　Pygame 的使用

本节主要讲解用 Pygame 开发游戏的逻辑、鼠标事件的处理、键盘事件的处理、字体的使用和声音的播放等基础知识，最后以一个“移动的坦克”例子来体现这些基础知识的应用。

17.2.1　Pygame 开发游戏的主要流程

Pygame 开发游戏的基础是创建游戏窗口，核心是处理事件、更新游戏状态和在屏幕上绘图。游戏状态可理解为程序中所有变量值的列表。在有些游戏中，游戏状态包括存放人物健康和位置的变量、物体或图形位置的变化，这些值可以在屏幕上显示。

物体或图形位置的变化只有通过在屏幕上绘图才能看出来。

可以简单地抽象出 Pygame 开发游戏的主要流程，如图 17-1 所示。

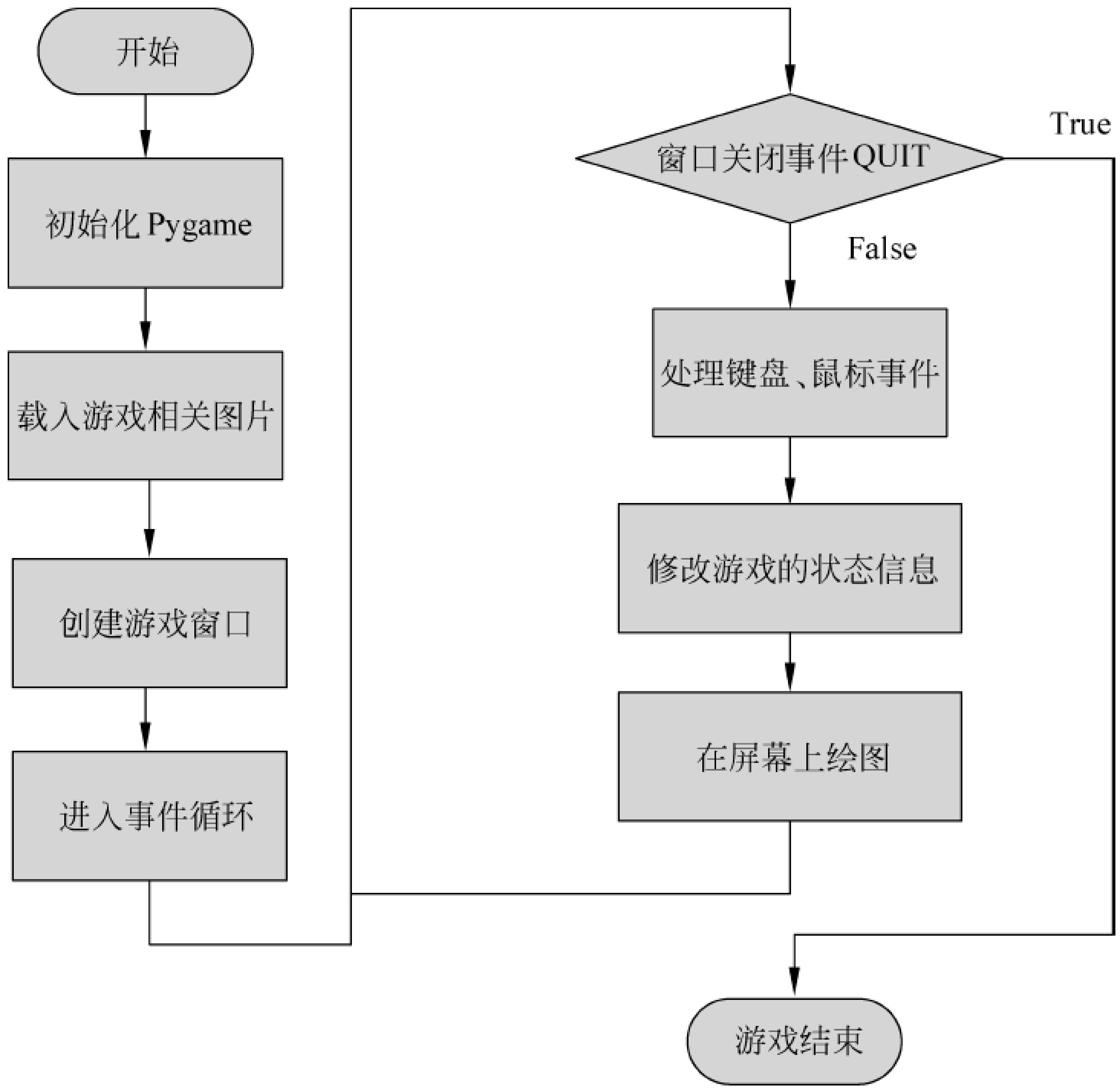


图 17-1　Pygame 开发游戏的主要流程

下面举一个具体例子来说明。

例 17-1　使用 Pygame 开发一个显示“Hello World!”标题的游戏窗口。

```
import pygame                                           #导入 pygame 模块
from pygame.locals import *
import sys
def hello world():
    pygame.init()     #任何 Pygame 程序均需要执行此语句进行模块的初始化
    #设置窗口的模式，(680,480)表示窗口像素
    #此函数返回一个 surface 对象，本程序不使用它，故没保存到对象变量中
    pygame.display.set mode((680,480))
    pygame.display.set_caption('Hello World!')         #设置窗口标题

    #无限循环，直到接收到窗口关闭事件
    while True:
        #处理事件
        for event in pygame.event.get():
            if event.type==QUIT:                        #接收到窗口关闭事件
                pygame.quit()                           #退出
                sys.exit()
        #将 surface 对象绘制在屏幕上
        pygame.display.update()
```

```
if __name__=="__main__":
    hello_world()
```

程序运行后仅见到黑色的游戏窗口，标题是"Hello World!"，如图 17-2 所示。

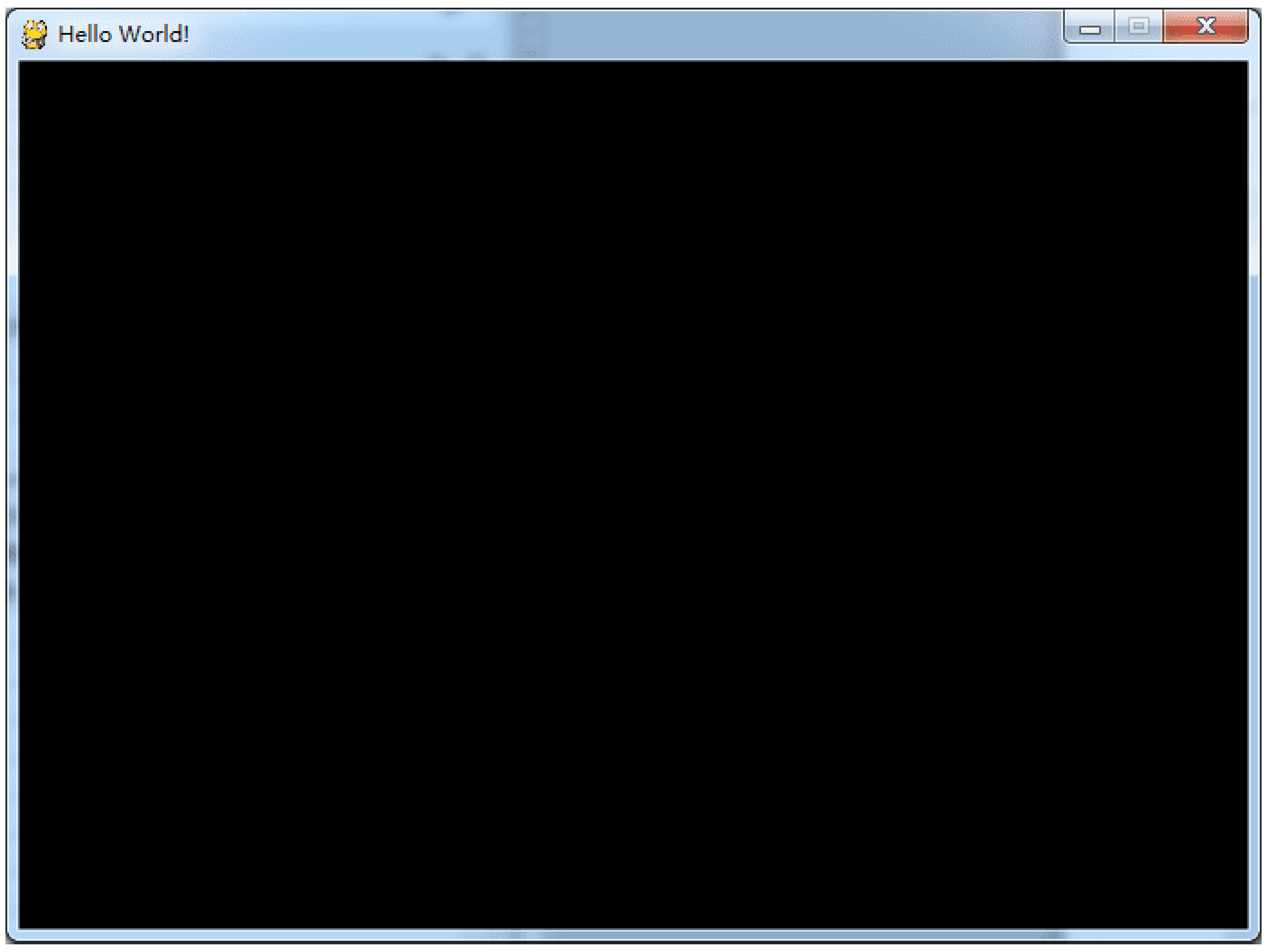


图 17-2　用 Pygame 开发的游戏窗口

在导入 Pygame 模块后，任何 Pygame 游戏程序均需要执行 pygame.init()语句进行模块的初始化，它必须在进入游戏的无限循环之前被调用。这个函数会自动初始化其他所有模块（例如 pygame.font 和 pygame.image），通过它载入驱动和硬件请求，这样游戏程序才可以使用计算机上的所有设备，比较费时间。如果只使用少量模块，应该分别初始化这些模块以节省时间，例如 pygame.sound.init()仅仅初始化声音模块。

该代码中有个无限循环，每个 Pygame 程序都需要它，在无限循环中可以做以下操作。

（1）处理事件：例如鼠标、键盘、关闭窗口等事件。

（2）更新游戏状态：例如坦克的位置变化、数量变化等。

（3）在屏幕上绘图：例如绘制新的敌方坦克等。

不断重复上面的 3 个步骤，从而完成游戏逻辑。

在本例代码中仅仅处理关闭窗口事件，也就是玩家关闭窗口时 pygame.quit()退出游戏。

17.2.2　Pygame 的图像/图形绘制

❶ Pygame 的图像绘制

Pygame 支持多种存储图像的方式（也就是图片格式），例如 JPEG、PNG 等，具体支持 JPEG（一般扩展名为.jpg 或者.jpeg，数码相机、网上的图片基本上都是这种格式，这是一种有损压缩方式，尽管对图片的质量有些损坏，但对于减小文件尺寸非常棒，其优点很多，只是不支持透明）、PNG（支持透明，无损压缩）、GIF（网上使用的很多，支持透明和动画，但只能有 256 种颜色，在软件和游戏中的使用很少）以及 BMP、PCX、TGA、TIF 等格式。

Pygame 使用 surface 对象来加载绘制的图像。对于 Pygame，加载图片使用 pygame.image.load()，给它一个文件名然后就返回一个 surface 对象。尽管读入的图像格式不同，但 surface 对象隐藏了这些不同。用户可以对一个 surface 对象进行涂画、变形、复制等各种操作。事实上，游戏屏幕也只是一个 surface，pygame.display.set_mode()返回了一个 surface 对象。

对于任何一个 surface 对象，可以用 get_width()、get_height()和 gei_size()函数来获取它的尺寸，get_rect()用来获取它的区域形状。

例 17-2 使用 Pygame 开发一个显示坦克自由移动的游戏窗口。

```
import pygame
from pygame.locals import *
import sys
def play tank():
    pygame.init()
    window size=(width, height)=(600, 400)          #窗口大小
    speed=[1, 1]              #坦克运行偏移量，即[水平,垂直]，值越大，移动越快
    color black=(255, 255, 255)                      #窗口背景色 RGB 值（白色）
    screen=pygame.display.set_mode(window_size)      #设置窗口模式
    pygame.display.set_caption('自由移动的坦克')      #设置窗口标题
    tank image=pygame.image.load('tankU.bmp')
                                         #加载坦克图片，返回一个 surface 对象
    tank rect=tank image.get rect()                  #获取坦克图片的区域形状
    while True:                                      #无限循环
        for event in pygame.event.get():
            if event.type==pygame.QUIT:              #退出事件处理
                pygame.quit()
                sys.exit()
        #使坦克移动，速度由 speed 变量控制
        tank rect=tank rect.move(speed)
        #当坦克运动出窗口时重新设置偏移量
        if(tank rect.left < 0) or (tank rect.right > width):    #水平方向
            speed[0]=-speed[0]                   #水平方向反向
        if(tank_rect.top < 0) or (tank_rect.bottom > height):   #垂直方向
            speed[1]=-speed[1]                   #垂直方向反向
        screen.fill(color_black)                 #填充窗口背景
        screen.blit(tank image, tank rect) #在窗口指定区域 tank rect 上绘制坦克
        pygame.display.update()                  #更新窗口显示内容
if __name__=='__main__':
    play_tank()
```

程序运行后，可以看到白色背景的游戏窗口，标题是“自由移动的坦克”，如图 17-3 所示。

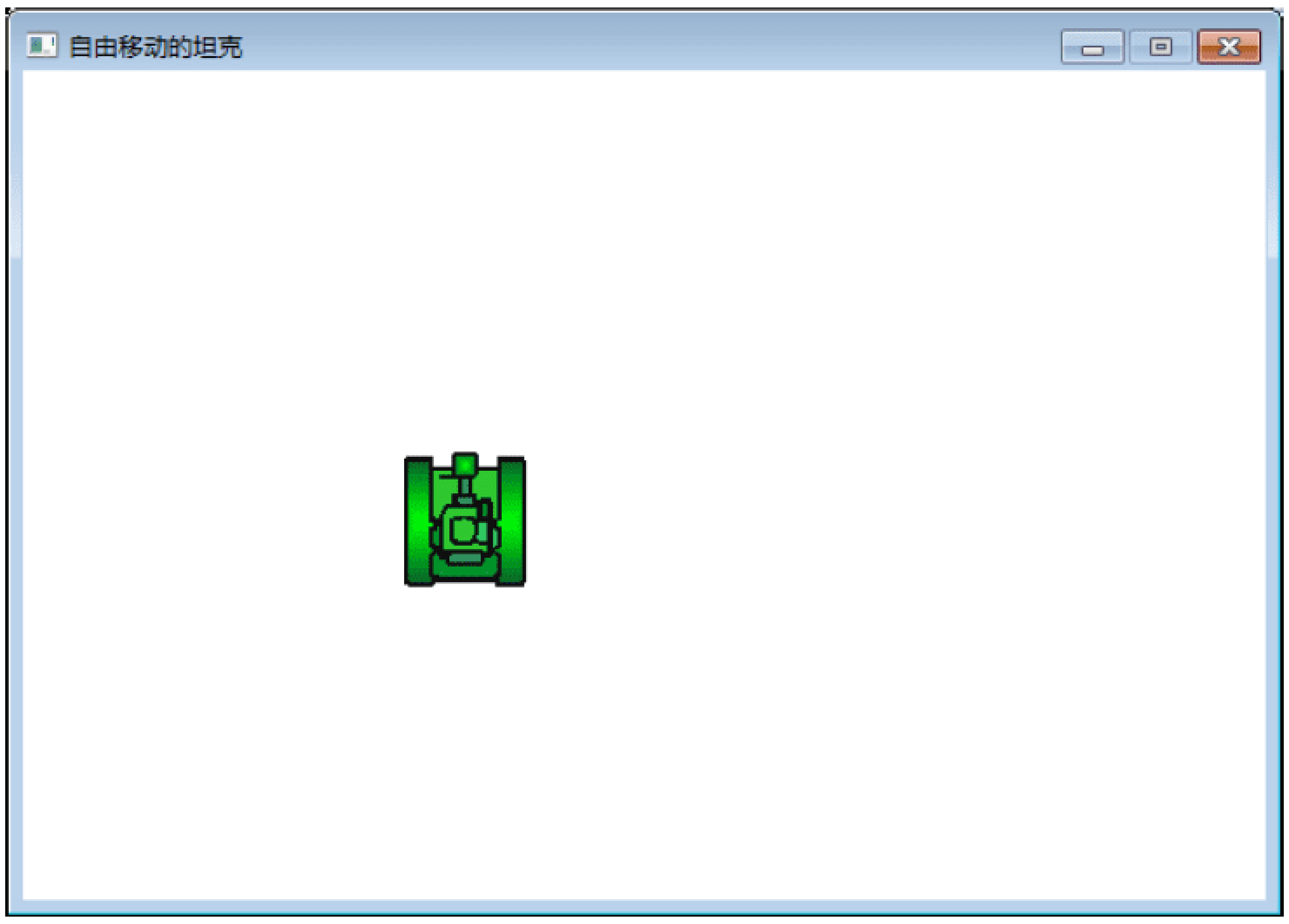


图 17-3　自由移动的坦克游戏窗口

在该游戏中通过修改坦克图像（surface 对象）区域的 left 属性（可以认为是 x 坐标）、surface 对象的 top 属性（可以认为是 y 坐标）改变坦克位置，从而显示出坦克自由移动的效果。在窗口（窗口也是 surface 对象）使用的 blit()函数上绘制坦克图像，最后注意需要更新窗口显示内容。

设置 fpsClock 变量的值即可控制游戏速度，语法如下：

```
fpsClock=pygame.time.Clock()
```

在无限循环中写入 fpsClock.tick(50)，可以按指定帧频 50 更新游戏画面（即每秒钟刷新 50 次屏幕）。

❷ Pygame 的图形绘制

在屏幕上绘制各种图形时使用 pygame.draw 模块中的一些函数，事实上 Pygame 可以不加载任何图片，而使用图形来制作一个游戏。

pygame.draw 中函数的第 1 个参数总是一个 surface，然后是颜色，接着是一系列的坐标等。对于计算机中的坐标，(0,0)代表左上角，水平向右为 X 轴的正方向，垂直向下为 Y 轴的正方向。该函数的返回值是一个 rect 对象，包含了绘制的区域，这样就可以很方便地更新那个部分了。pygame.draw 中的函数如表 17-3 所示。

表 17-3　pygame.draw 中的函数

函　数	作　用	函　数	作　用
rect()	绘制矩形	line()	绘制线
polygon()	绘制多边形（3 个及 3 个以上的边）	lines()	绘制一系列的线
circle()	绘制圆	aaline()	绘制一根平滑的线
ellipse()	绘制椭圆	aalines()	绘制一系列平滑的线
arc()	绘制圆弧		

下面详细说明 pygame.draw 中各个函数的使用。

1）pygame.draw.rect()

格式：pygame.draw.rect(surface, color,rect,width=0)

pygame.draw.rect()在 surface 上画一个矩形，除了 surface 和 color 以外，rect 接受一个矩形的坐标和线宽参数，如果线宽是 0 或省略，则填充。

2）pygame.draw.polygon()

格式：pygame.draw.polygon(surface, color, pointlist, width=0)

polygon()用于画多边形，其用法类似于 rect()，与 rect()的第 1、第 2、第 4 个参数都是相同的，只不过 polygon()会接受一系列坐标的列表，代表了各个顶点的坐标。

3）pygame.draw.circle()

格式：pygame.draw.circle(surface, color, pos, radius, width=0)

circle()用于画一个圆，它接受一个圆心坐标和半径参数。

4）pygame.draw.ellipse()

格式：pygame.draw.ellipse(surface, color, rect, width=0)

用户可以把 ellipse 想象成一个被压扁的圆，事实上，它可以被一个矩形装起来。pygame.draw.ellipse()的第 3 个参数就是这个椭圆的外接矩形。

5）pygame.draw.arc()

格式：pygame.draw.arc(surface,color,rect,start_angle,stop_angle,width=1)

arc 是椭圆的一部分，所以它的参数比椭圆多一点。但是它是不封闭的，因此没有 fill()方法。start_angle 和 stop_angle 为开始和结束的角度。

6）pygame.draw.line()

格式：pygame.draw.line(surface,color,start_pos,end_pos,width=1)

line()用于画一条线段，start_pos、end_pos 是线段起点和终点坐标。

7）pygame.draw.lines()

格式：pygame.draw.lines(surface,color,closed,pointlist,width=1)

closed 是一个布尔变量，指明是否需要多画一条线来使这些线条闭合（这样就和 polygon 一样了），pointlist 是一个顶点坐标的数组。

17.2.3 Pygame 的键盘和鼠标事件的处理

所谓事件，就是程序上发生的事。例如用户按键盘上的某个键或是单击、移动鼠标。对于这些事件，游戏程序需要做出反应。在例 17-2 中，程序会一直运行下去，直到用户关闭窗口而产生了一个 QUIT 事件，Pygame 会接收用户的各种操作（例如按键盘上的键、移动鼠标等）产生事件。事件随时可能发生，而且量可能会很大，Pygame 的做法是把一系列事件存放到一个队列里逐个处理。

在例 17-2 中使用了 pygame.event.get()来处理所有事件，如果使用 pygame.event.wait()，Pygame 会等到发生一个事件时才继续下去，一般在游戏中不太实用，因为游戏往往是需要动态运作的。Pygame 中的常用事件如表 17-4 所示。

表 17-4　Pygame 中的常用事件

事　　件	产 生 途 径	参　　数
QUIT	用户按下“关闭”按钮	none
ACTIVEEVENT	Pygame 被激活或者隐藏	gain、state
KEYDOWN	键盘被按下	unicode、key、mod
KEYUP	键盘被放开	key、mod
MOUSEMOTION	鼠标移动	pos、rel、buttons
MOUSEBUTTONDOWN	鼠标被按下	pos、button
MOUSEBUTTONUP	鼠标被放开	pos、button

❶ Pygame 的键盘事件的处理

通常用 pygame.event.get()获取所有事件，若 event.type == KEYDOWN，这时是键盘事件，再判断按键 event.key 的种类（即 K_a、K_b、K_LEFT 这种形式）。用户也可以使用 pygame.key.get_pressed()获取所有被按下的键值，它会返回一个元组。这个元组的索引就是键值，对应的就是键是否被按下。

```
pressed_keys=pygame.key.get_pressed()
if pressed_keys[K_SPACE]:
    #空格键被按下
    fire()#发射子弹
```

在 key 模块下有很多函数。

- key.get_focused()：返回当前的 Pygame 窗口是否被激活。
- key.get_pressed()：获得所有按下的键值。
- key.get_mods()：按下的组合键（Alt、Ctrl、Shift）。
- key.set_mods()：模拟按下组合键的效果（KMOD_ALT、KMOD_CTRL、KMOD_SHIFT）。

例 17-3　使用 Pygame 开发一个由用户控制坦克移动的游戏。在例 17-2 的基础上增加通过方向键控制坦克运动的功能，并为游戏增加背景图片。程序的运行效果如图 17-4 所示。

```
import os
import sys
import pygame
from pygame.locals import *
def control_tank(event):                    #控制坦克运动函数
    speed=[x,y]=[0, 0]          #相对坐标
    speed_offset=1                          #速度
    #当方向键被按下时进行位置计算
    if event.type==pygame.KEYDOWN:
        if event.key==pygame.K_LEFT:
            speed[0]-=speed_offset
        if event.key==pygame.K_RIGHT:
            speed[0]=speed_offset
```

```
            if event.key==pygame.K_UP:
                speed[1]-=speed offset
            if event.key==pygame.K DOWN:
                speed[1]=speed offset
        #当方向键被释放时相对偏移为 0，即不移动
        if event.type in (pygame.KEYUP, pygame.K LEFT, pygame.K RIGHT, pygame
        .K DOWN) :
            speed=[0, 0]
        return speed
    def play_tank():
        pygame.init()
        window_size=Rect(0, 0, 600, 400)                    #窗口大小
        speed=[1, 1]              #坦克运行偏移量，即[水平,垂直]，值越大，移动越快
        color_black=(255, 255, 255)        #窗口背景色 RGB 值（白色）
        screen=pygame.display.set mode(window size.size)    #设置窗口模式
        pygame.display.set_caption('用户方向键控制坦克移动')   #设置窗口标题
        tank_image=pygame.image.load('tankU.bmp')           #加载坦克图片
        #加载窗口背景图片
        back image=pygame.image.load('back image.jpg')
        tank rect=tank image.get rect()        #获取坦克图片的区域形状
        while True:
            #退出事件处理
            for event in pygame.event.get():    #pygame.event.get()获取事件序列
                if event.type==pygame.QUIT:
                    pygame.quit()
                    sys.exit()
            #使坦克移动，速度由 speed 变量控制
            cur speed=control tank(event)
            #rect 的 clamp()方法使移动范围限制在窗口内
            tank rect=tank rect.move(cur speed).clamp(window size)
            screen.blit(back image,(0, 0))                  #设置窗口背景图片
            screen.blit(tank_image,tank_rect)               #在窗口上绘制坦克
            pygame.display.update()                         #更新窗口显示内容
    if __name__=='__main__':
        play_tank()
```

当用户按下方向键，计算出相对位置 cur_speed 后，使用 tank_rect.move(cur_speed)函数向指定方向移动坦克。当释放方向键时坦克停止移动。

❷ Pygame 的鼠标事件的处理

pygame.mouse 的函数如下。

- pygame.mouse.get_pressed()：返回按键的按下情况，返回的是一元组，分别为左键、中键、右键，如果被按下则为 True。

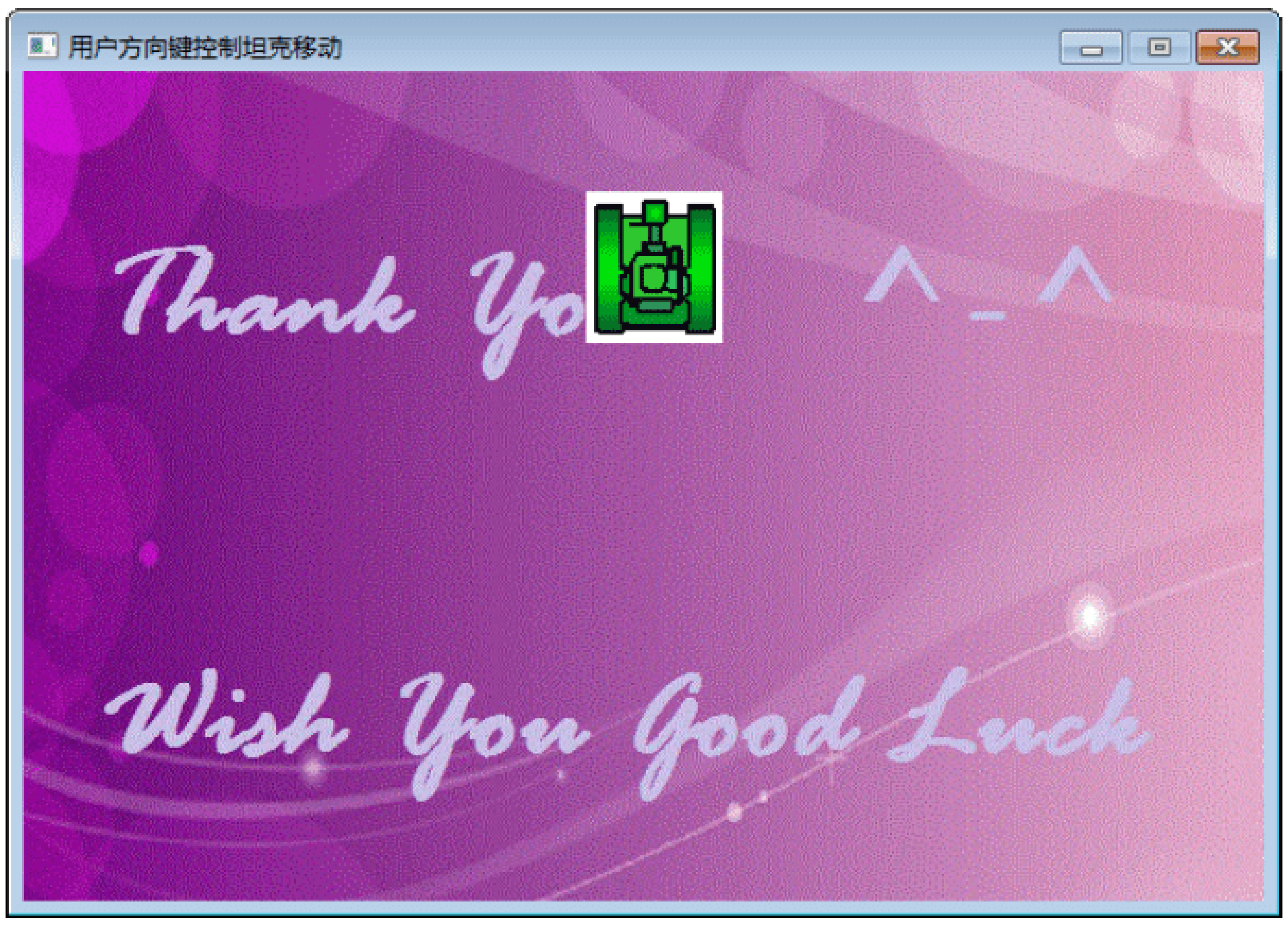


图 17-4　用方向键控制坦克运动的游戏窗口

- pygame.mouse.get_rel()：返回相对偏移量，即（x 方向偏移量，y 方向偏移量）的一元组。
- pygame.mouse.get_pos()：返回当前鼠标位置(x, y)。

例如“x, y = pygame.mouse.get_pos()”用于获取鼠标位置。

- pygame.mouse.set_pos()：设置鼠标位置。
- pygame.mouse.set_visible()：设置鼠标光标是否可见。
- pygame.mouse.get_focused()：如果鼠标在 Pygame 窗口内有效，返回 True。
- pygame.mouse.set_cursor()：设置鼠标的默认光标样式。
- pygame.mouse.get_cursor()：返回鼠标的光标样式。

例 17-4　演示鼠标事件处理的程序，运行效果如图 17-5 所示。

```
import pygame
from pygame.locals import *
from sys import exit
from random import *
from math import pi
pygame.init()
screen=pygame.display.set mode((640, 480), 0, 32)
points=[]
while True:
    for event in pygame.event.get():
        if event.type==QUIT:
            pygame.quit()
            exit()
        if event.type==KEYDOWN:
            #按任意键可以清屏，并把点回复到原始状态
            points=[]
            screen.fill((255,255,255))                    #用白色填充窗口背景
        if event.type==MOUSEBUTTONDOWN:                   #鼠标按下
```

```
        screen.fill((255,255,255))
        #画随机矩形
        rc=(255,0,0)                        #红色
        rp=(randint(0,639), randint(0,479))
        rs=(639-randint(rp[0],639),479-randint(rp[1],479))
        pygame.draw.rect(screen, rc,Rect(rp,rs))
        #画随机圆形
        rc=(0,255,0)                        #绿色
        rp=(randint(0,639),randint(0,479))
        rr=randint(1,200)
        pygame.draw.circle(screen,rc,rp,rr)
        #获得当前鼠标单击位置
        x,y=pygame.mouse.get_pos()
        points.append((x,y))
        #根据单击位置画弧线
        angle=(x/639.)*pi*2.
        pygame.draw.arc(screen,(0,0,0),(0,0,639,479),0,angle,3)
        #根据单击位置画椭圆
        pygame.draw.ellipse(screen,(0,255,0),(0,0,x,y))
        #从左上和右下画两根线连接到单击位置
        pygame.draw.line(screen,(0,0,255),(0,0),(x,y))
        pygame.draw.line(screen,(255,0,0),(640,480),(x,y))
        #画单击轨迹图
        if len(points)>1:
            pygame.draw.lines(screen, (155,155,0),False,points,2)
        #和轨迹图基本一样，只不过是闭合的，因为会覆盖，所以这里注释了
        #if len(points)>=3:
        #   pygame.draw.polygon(screen,(0,155,155),points,2)
        #把每个点画明显一点
        for p in points:
            pygame.draw.circle(screen,(155,155,155),p,3)
    pygame.display.update()
```

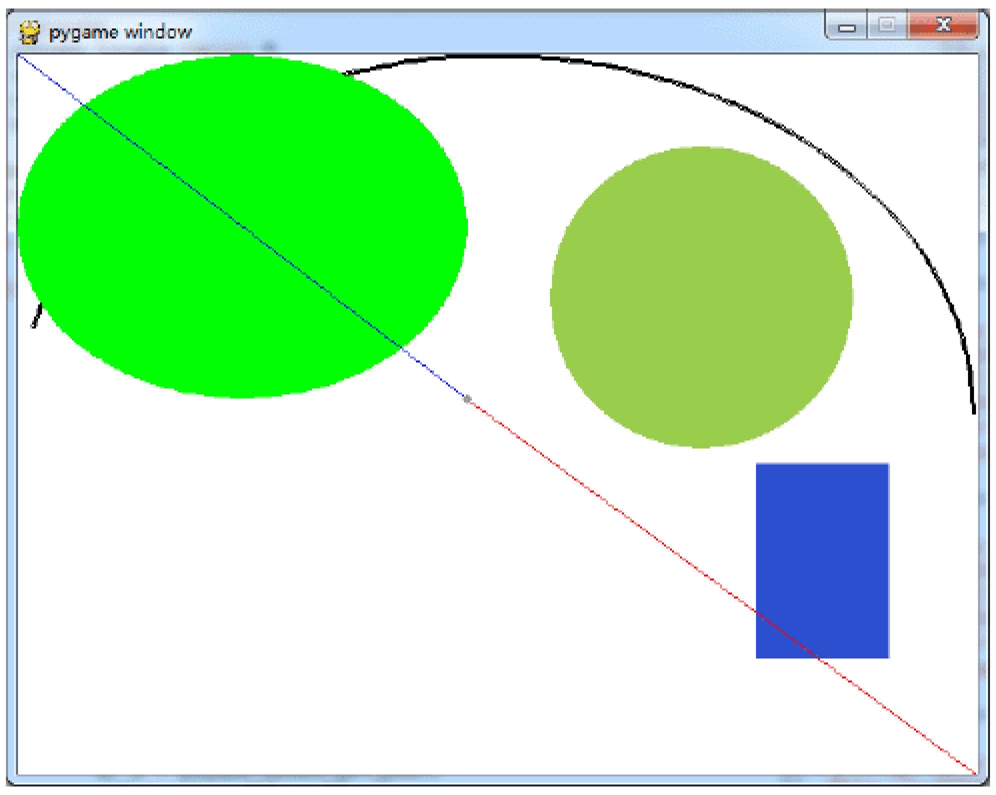


图 17-5　演示鼠标事件处理的程序的运行效果

运行这个程序，在窗口上单击鼠标就会有图形出来，按任意键可以重新开始。

17.2.4　Pygame 的字体使用

Pygame 可以直接调用系统字体，也可以调用 TTF 字体。为了使用字体，首先应该创建一个 Font 对象，对于系统自带的字体应该这样调用：

```
font1=pygame.font.SysFont('arial', 16)
```

第 1 个参数是字体名，第 2 个参数是字号。在正常情况下系统里都会有 arial 字体，如果没有会使用默认字体，默认字体和用户使用的系统有关。

用户可以使用 pygame.font.get_fonts()来获取当前系统所有的可用字体：

```
>>> pygame.font.get fonts()
'gisha', 'fzshuti', 'simsunnsimsun', 'estrangeloedessa',
'symboltigerexpert', 'juiceitc', 'onyx', 'tiger', 'webdings',
'franklingothicmediumcond', 'edwardianscriptitc'
```

另外还有一种调用方法是使用自己的 TTF 字体：

```
my_font=pygame.font.Font("my_font.ttf", 16)
```

这个方法的好处是可以把字体文件和游戏一起打包分发，避免玩家计算机上没有这个字体而无法显示的问题。一旦有了 Font 对象，就可以用 render()方法来设置文字内容，然后通过 blit()方法写到屏幕上：

```
text=font1.render("坦克大战",True,(0,0,0),(255,255,255))
```

render()方法的第 1 个参数是写入的文字内容；第 2 个参数是布尔值，说明是否开启抗锯齿；第 3 个参数是字体本身的颜色；第 4 个参数是背景的颜色。如果不想有背景色，也就是让背景透明，可以不加第 4 个参数。

例如自己定义一个文字处理函数 show_text()，其中参数 surface_handle 为 surface 句柄，pos 为文字显示位置，color 为文字颜色，font_bold 为是否加粗，font_size 为字体大小，font_italic 为是否斜体。

```
def show text(surface handle, pos, text, color, font bold=False, font
size=13, font italic=False):
    #cur_font=pygame.font.SysFont("宋体", font_size)     #获取系统字体
    cur_font=pygame.font.Font('simfang.ttf', 30)     #获取字体，并设置文字大小
    cur font.set bold(font bold)                     #设置是否加粗
    cur font.set italic(font italic)                 #设置是否斜体
    text fmt=cur font.render(text, 1, color)         #设置文字内容
    surface_handle.blit(text_fmt, pos)               #绘制文字
```

在更新窗口内容 pygame.display.update()之前加入：

```
text_pos=u "坦克大战"
show text(screen, (20, 220), text pos, (255, 0, 0), True)
text_pos=u"坦克位置:(%d,%d)" % (tank_rect.left, tank_rect.top)
```

```
show_text(screen, (20, 420), text_pos, (0, 255, 255), True)
```

此时会在屏幕上的(20, 220)处显示红色的“坦克大战”文字，并且在(20, 420)处显示现在坦克所处位置的坐标，移动坦克，位置坐标文字同时会改变。

17.2.5　Pygame 的声音播放

❶ Sound 对象

在初始化声音设备后就可以读取一个音乐文件到一个 Sound 对象中。pygame.mixer.sound()接受一个文件名，也可以是一个文件对象，不过这个文件必须是 WAV 或者 OGG 文件。

```
hello_sound=pygame.mixer.sound("hello.ogg")     #建立 Sound 对象
hello_sound.play()                              #声音播放一次
```

一旦这个 Sound 对象出来了，就可以使用 play()来播放它。play(loop, maxtime)可以接受两个参数，loop 是重复的次数（取 1 是两次，注意是重复的次数而不是播放的次数），-1 意味着无限循环；maxtime 是指多少毫秒后结束。

若不使用任何参数调用，意味着把这个声音播放一次。一旦 play()方法调用成功，就会返回一个 Channel 对象，否则返回一个 None。

❷ music 对象

在 Pygame 中还提供了 pygame.mixer.music 类来控制背景音乐的播放。pygame.mixer.music 用来播放 MP3 和 OGG 音乐文件，不过 MP3 并不是所有的系统都支持（Linux 默认就不支持 MP3 播放）。用户可以用 pygame.mixer.music.load()加载一个文件，然后使用 pygame.mixer.music.play()播放，不放的时候就用 stop()方法停止，当然也有类似录影机上的 pause()和 unpause()方法。

```
#加载背景音乐
pygame.mixer.music.load("hello.mp3")
pygame.mixer.music.set volume(music volume/100.0)
#循环播放，从音乐的第 30 秒开始
pygame.mixer.music.play(-1, 30.0)
```

在游戏退出事件中加入停止音乐播放的代码：

```
#停止音乐播放
pygame.mixer.music.stop()
```

music 对象提供了丰富的函数方法，下面分别介绍。

1）pygame.mixer.music.load()

功能：加载音乐文件。

格式：pygame.mixer.music.load(filename)

2）pygame.mixer.music.play()

功能：播放音乐。

格式：pygame.mixer.music.play(loops=0,start=0.0)

其中，loops 表示循环次数，如果设置为-1，表示不停地循环播放，如果 loops 为 5，则播放 5+1=6 次；start 参数表示从音乐文件的哪一秒开始播放，设置为 0 表示从开始完整播放。

3）pygame.mixer.music.rewind()

功能：重新播放。

格式：pygame.mixer.music.rewind()

4）pygame.mixer.music.stop()

功能：停止播放。

格式：pygame.mixer.music.stop()

5）pygame.mixer.music.pause()

功能：暂停播放。

格式：pygame.mixer.music.pause()

用户可通过 pygame.mixer.music.unpause()恢复播放。

6）pygame.mixer.music.set_volume()

功能：设置音量。

格式：pygame.mixer.music.set_volume(value)

其中，value 的取值为 0.0～1.0。

7）pygame.mixer.music.get_pos()

功能：获取当前播放了多长时间。

格式：pygame.mixer.music.get_pos(): return time

17.2.6　Pygame 的精灵使用

pygame.sprite.Sprite 是 Pygame 中用来实现精灵的一个类，在使用时并不需要对它实例化，只需要继承它，然后按需写出自己的类，因此非常简单、实用。

❶ 精灵

精灵可以被认为是一个个小图片（帧）序列（例如人物行走），它可以在屏幕上移动，并且可以与其他图形对象交互。精灵图像可以是使用 Pygame 绘制形状函数绘制的形状，也可以是图像文件。图 17-6 所示为由 16 帧图片组成的人物行走序列。

❷ Sprite 类的成员

pygame.sprite.Sprite 用来实现精灵类，Sprite 的数据成员和函数方法主要如下。

1）self.image

其负责显示什么图形，例如 self.image=pygame.Surface([x,y])说明该精灵是一个 x×y 大小的矩形，self.image=pygame.image.load(filename)说明该精灵显示 filename 这个图片文件。

self.image.fill([color])负责对 self.image 着色，例如：

```
self.image=pygame.Surface([x,y])
self.image.fill([255,0,0])          #对 x×y 大小的矩形填充红色
```

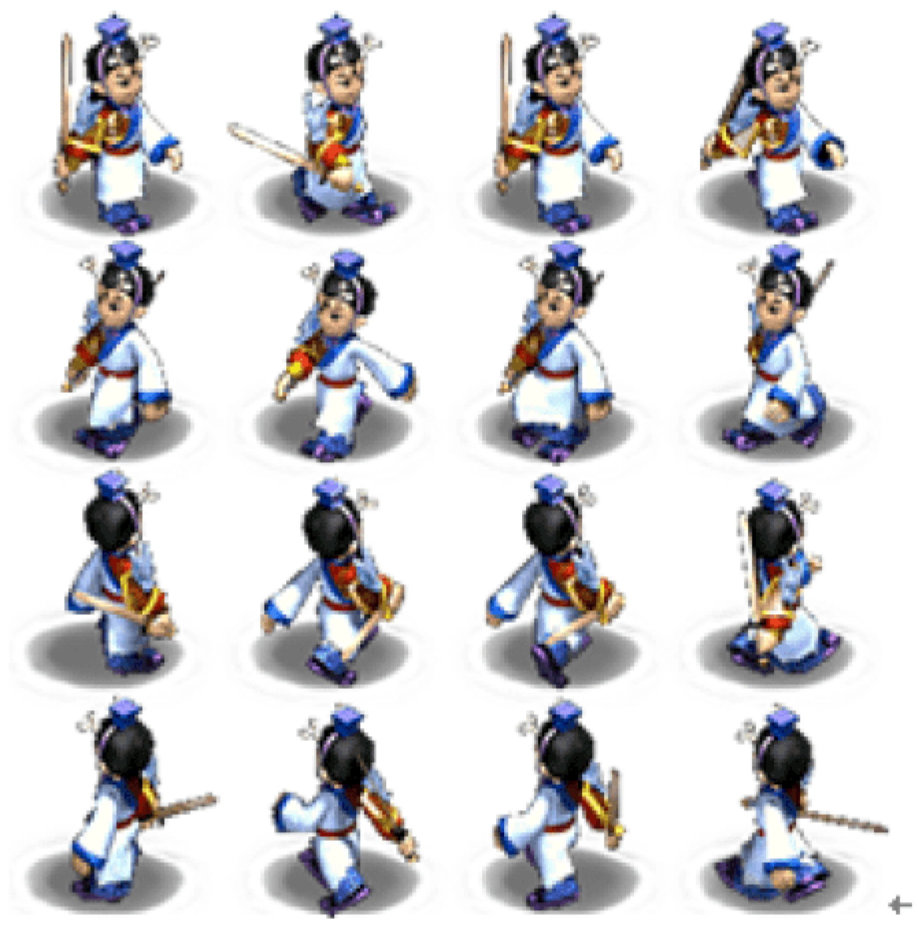


图 17-6　精灵图片序列

2）self.rect

其负责在哪里显示。一般来说，先用 self.rect=self.image.get_rect()获取 image 矩形大小，然后给 self.rect 设定显示的位置，一般用 self.rect.topleft 确定左上角的显示位置，当然也可以用 topright、bottomright、bottomleft 分别确定其他几个角的位置。

另外，self.rect.top、self.rect.bottom、self.rect.left、self.rect.right 分别表示上、下、左、右。

3）self.update()

其负责使精灵行为生效。

4）Sprite.add()

添加精灵到 groups 中。

5）Sprite.remove()

从精灵组 groups 中删除。

6）Sprite.kill()

从精灵组 groups 中删除全部精灵。

7）Sprite.alive()

判断某个精灵是否属于精灵组 groups。

❸ 建立精灵

所有精灵在建立时都是从 pygame.sprite.Sprite 中继承的，建立精灵要设计自己的精灵类。

例 17-5　建立 Tank 精灵。

```
import pygame,sys
pygame.init()
class Tank(pygame.sprite.Sprite):
    def __init__(self,filename,initial position):
        pygame.sprite.Sprite.__init__(self)
        self.image=pygame.image.load(filename)
        self.rect=self.image.get_rect()              #获取 self.image 的大小
        #self.rect.topleft=initial_position          #确定左上角的显示位置
```

```
        self.rect.bottomright=initial_position  #右下角的显示位置是[150,100]

screen=pygame.display.set mode([640,480])
screen.fill([255,255,255])
fi='tankU.jpg'
b=Tank(fi,[150,100])
while True:
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            sys.exit()
    screen.blit(b.image,b.rect)
    pygame.display.update()
```

例 17-6　使用图 17-6 所示的精灵图片序列建立动画效果的人物行走精灵。

在游戏动画中，人物行走是基本动画，在精灵中不断切换人物行走图片，从而达到动画的效果。

```
import pygame
from pygame.locals import *
class MySprite(pygame.sprite.Sprite):
    def __init__(self, target):
        pygame.sprite.Sprite.__init__(self)
        self.target surface=target
        self.image=None
        self.master image=None
        self.rect=None
        self.topleft=0,0
        self.frame=0
        self.old frame=-1
        self.frame width=1
        self.frame height=1
        self.first_frame=0          #第 1 帧序号
        self.last frame=0           #最后一帧序号
        self.columns=1              #列数
        self.last_time=0
```

在加载一个精灵图片序列的时候需要告知程序一帧的大小（传入帧的宽度和高度以及文件名和列数）。

```
    def load(self, filename, width, height, columns):
        self.master image=pygame.image.load(filename).convert alpha()
        self.frame width=width
        self.frame height=height
        self.rect=0,0,width,height
        self.columns=columns
        rect=self.master_image.get_rect()
```

```
    self.last_frame=(rect.width//width)*(rect.height//height)-1
```

一个循环动画通常是这样工作的：从第 1 帧开始不断地加载直到最后一帧，然后再返回第 1 帧，并不断重复这个操作。

但是如果只是这样做，程序会一股脑地将动画播放完，这里想让它根据时间间隔一张一张地播放，因此加入定时的代码，将帧速率 ticks 传递给 Sprite 的 update()函数，这样就可以轻松地让动画按照帧速率来播放。

```
    def update(self, current_time, rate=60):
        if current_time>self.last_time + rate: #如果时间超过上次时间+60 毫秒
            self.frame+=1                        #帧号加 1，意味着显示下一帧图像
        if self.frame>self.last_frame:           #帧号超过最后一帧
            self.frame=self.first_frame          #回到第 1 帧
            self.last_time=current_time
        if self.frame!=self.old_frame:
            #首先需要计算单个帧左上角的 x、y 位置值
            frame_x=(self.frame % self.columns) * self.frame_width
            frame_y=(self.frame // self.columns) * self.frame_height
            #然后将计算好的 x,y 值传递给位置属性
            rect=(frame_x, frame_y, self.frame_width, self.frame_height)
                                                 #要显示区域
            self.image=self.master_image.subsurface(rect)
                                                 #截取要显示区域图像
            self.old_frame=self.frame
pygame.init()
screen=pygame.display.set_mode((800,600),0,32)
pygame.display.set_caption("精灵类测试")
font=pygame.font.Font(None, 18)
#启动一个定时器，然后调用 tick(num)函数就可以让游戏以 num 帧来运行
framerate=pygame.time.Clock()
cat=MySprite(screen)
cat.load("sprite2.png", 92, 95, 4)     #精灵图片，每帧 92×95 大小，共 4 列
group=pygame.sprite.Group()
group.add(cat)
while True:
    framerate.tick(10)                 #指定帧速率
    ticks=pygame.time.get_ticks()      #获取运行时间
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            pygame.quit()
            exit()
        key=pygame.key.get_pressed()
        if key[pygame.K_ESCAPE]:       #Esc 键
            exit()
```

```
        screen.fill((0,0,100))
        #cat.draw(screen)                                          #没有此方法
        cat.update(ticks)
        screen.blit(cat.image,cat.rect)
        #group.update(ticks)
        #group.draw(screen)
        pygame.display.update()
```

运行后可见一个人物行走动画，用户也可以使用精灵组的 update()和 draw()函数实现精灵动画。

```
    group.update(ticks)
                #将帧速率 ticks 传递给 sprite 的 update()函数，让动画按照帧速率来播放
    group.draw(screen)
```

❹ 建立精灵组

当程序中有大量实体的时候，操作这些实体将会是一件相当麻烦的事，那么有没有什么容器可以将这些精灵放在一起统一管理呢？答案就是使用精灵组。

Pygame 使用精灵组来管理精灵的绘制和更新，精灵组是一个简单的容器。

使用 pygame.sprite.Group()函数可以创建一个精灵组：

```
    group=pygame.sprite.Group()
    group.add(sprite_one)
```

精灵组也有 update()和 draw()函数：

```
    group.update()
    group.draw()
```

Pygame 还提供了精灵与精灵之间的冲突检测、精灵与组之间的碰撞检测，这些碰撞检测技术在 17.4 节的飞机大战游戏中要使用。

❺ 精灵与精灵之间的碰撞检测

1）两个精灵之间的矩形检测

在只有两个精灵的时候可以使用 pygame.sprite.collide_rect()函数进行一对一的冲突检测。这个函数需要传递两个精灵，并且每个精灵都需要继承自 pygame.sprite.Sprite。

这里举个例子：

```
    spirte_1=MySprite("sprite_1.png",200,200,1)
                                           #MySprite 是例 17-6 创建的精灵类
    sprite_2=MySprite("sprite_2.png",50,50,1)
    result=pygame.sprite.collide_rect(sprite_1,sprite_2)
    if result:
        print("精灵碰撞上了")
```

2）两个精灵之间的圆检测

矩形冲突检测并不适用于所有形状的精灵，因此在 Pygame 中还提供了圆形冲突检测。pygame.sprite.collide_circle()函数是基于每个精灵的半径值来进行检测的，用户可以自己指

定精灵半径，或者让函数计算精灵半径。

```
result=pygame.sprite.collide circle(sprite 1,sprite 2)
if result:
    print("精灵碰撞上了")
```

3）两个精灵之间的像素遮罩检测

如果矩形检测和圆形检测都不能满足需求，Pygame 还为用户提供了一个更加精确的检测——pygame.sprite.collide_mask()。

这个函数接受两个精灵作为参数，返回值是一个 bool 变量。

```
if pygame.sprite.collide mask(sprite 1,sprite 2):
    print("精灵碰撞上了")
```

4）精灵和组之间的矩形冲突检测

在调用 pygame.sprite.spritecollide(sprite,sprite_group,bool)函数的时候，一个组中的所有精灵都会逐个地对另外一个单个精灵进行冲突检测，发生冲突的精灵会作为一个列表返回。

这个函数的第 1 个参数是单个精灵，第 2 个参数是精灵组，第 3 个参数是一个 bool 值，最后这个参数起了很大的作用，当为 True 的时候会删除组中所有冲突的精灵，当为 False 的时候不会删除冲突的精灵。

```
list_collide=pygame.sprite.spritecollide(sprite,sprite_group,False);
```

另外，这个函数还有一个变体——pygame.sprite.spritecollideany()，这个函数在判断精灵组和单个精灵冲突的时候会返回一个 bool 值。

5）精灵组之间的矩形冲突检测

利用 pygame.sprite.groupcollide()函数可以检测两个组之间的冲突，它返回一个字典（键值对）。

以上学习了几种常用的冲突检测函数，下面在游戏实例中实际运用这些函数。

17.3 基于 Pygame 设计贪吃蛇游戏

贪吃蛇游戏通过玩家控制蛇移动，不断吃到食物（红色草莓）增长，直到蛇身碰到边界游戏结束。其运行效果如图 17-7 所示。

```
import pygame, sys, time, random
from pygame.locals import *
```

输入下面的两行来启用 Pygame，这样 Pygame 在该程序中就可以用了：

```
pygame.init()
fpsClock=pygame.time.Clock()
```

第 1 行告诉 Pygame 初始化，第 2 行创建一个名为 fpsClock 的变量，该变量用来控制游戏的速度。然后用下面的两行代码新建一个 Pygame 显示层（游戏元素画布）。

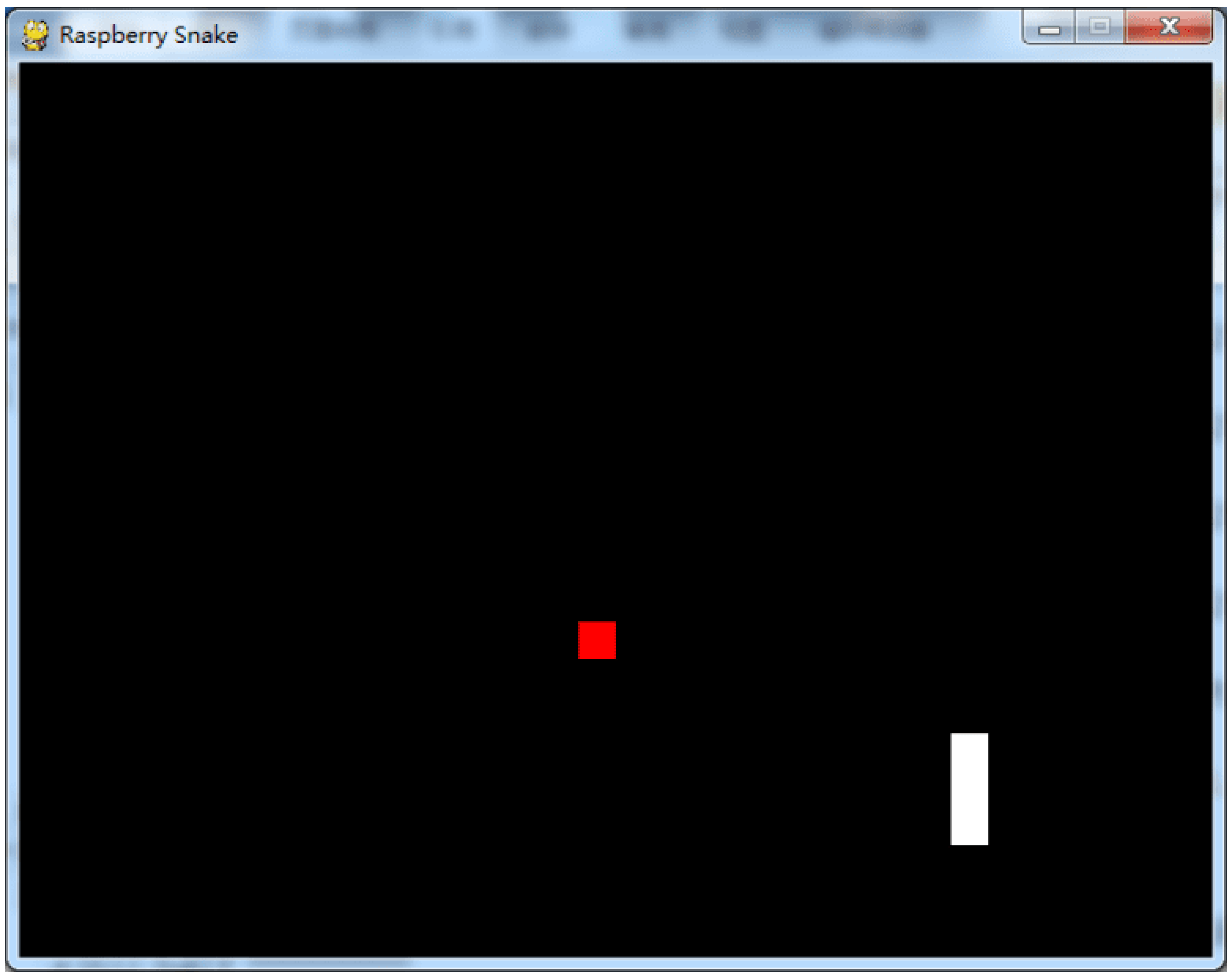


图 17-7　基于 Pygame 设计的贪吃蛇游戏的运行效果

```
playSurface=pygame.display.set mode((640, 480))
pygame.display.set_caption('Raspberry Snake')
```

接下来定义一些颜色，虽然这一步并不是必需的，但它会减少代码量。下面的代码定义了程序中用到的颜色：

```
redColour=pygame.Color(255, 0, 0)
blackColour=pygame.Color(0, 0, 0)
whiteColour=pygame.Color(255, 255, 255)
greyColour=pygame.Color(150, 150, 150)
```

下面的几行代码初始化了程序中用到的一些变量，这是很重要的一步，因为如果游戏开始时这些变量为空，Python 将无法正常运行。

```
snakePosition=[100,100]                          #蛇头位置
snakeSegments=[[100,100],[80,100],[60,100]]      #蛇身序列
raspberryPosition=[300,300]                      #草莓位置
raspberrySpawned=1                       #是否吃到草莓，1 为没有吃到，0 为吃到
direction='right'                                #运动方向，初始向右
changeDirection=direction
```

此时可以看到变量 snakePosition、snakeSegments 和 raspberryPosition 被设置为用逗号分隔的列表。

用下面几行代码来定义 gameOver()函数：

```
def gameOver():
        gameOverFont=pygame.font.Font('freesansbold.ttf', 72)
        gameOverSurf=gameOverFont.render('Game Over', True, greyColour)
```

```
            gameOverRect=gameOverSurf.get rect()
            gameOverRect.midtop=(320, 10)
            playSurface.blit(gameOverSurf, gameOverRect)
            pygame.display.flip()
            time.sleep(5)
            pygame.quit()
            sys.exit()
```

gameOver()函数用了一些 Pygame 命令来完成一个简单的任务：用大号字体将 Game Over 打印在屏幕上，停留 5 秒钟，然后退出 Pygame 和 Python 程序。在游戏开始之前就定义了结束函数，这看起来有点奇怪，但是所有的函数都应该在被调用前定义。Python 是不会自己执行 gameOver()函数的，直到用户调用该函数。

至此程序的开头部分已经完成，接下来进入主要部分。该程序运行在一个无限循环（一个永不退出的 while 循环）中，直到蛇撞到了墙或者自己才会结束游戏。首先用下面的代码开始主循环：

```
while True:
```

没有其他的比较条件，Python 会检测 True 是否为真。如果 True 一直为真，循环会一直进行，直到用户调用 gameOver()函数告诉 Python 退出该循环。

```
for event in pygame.event.get():
    if event.type==QUIT:
        pygame.quit()
        sys.exit()
    elif event.type==KEYDOWN:
        elif event.type==KEYDOWN:
            if event.key==K RIGHT or event.key==ord('d'):
                changeDirection='right'
            if event.key==K LEFT or event.key==ord('a'):
                changeDirection='left'
            if event.key==K UP or event.key==ord('w'):
                changeDirection='up'
            if event.key==K DOWN or event.key==ord('s'):
                changeDirection='down'
            if event.key==K ESCAPE:
                pygame.event.post(pygame.event.Event(QUIT))
```

for 循环用来检测按键等 Pygame 事件。

第 1 个检测 if event.type == QUIT 告诉 Python 如果 Pygame 发出了 QUIT 信息（当用户按下 Esc 键时），执行下面缩进的代码。之后的两行类似 gameOver()函数，通知 Pygame 和 Python 程序结束并退出。

第 2 个检测以 elif 开头的行用来检测 Pygame 是否发出 KEYDOWN 事件，该事件在用户按下键盘时产生。

KEYDOWN 事件修改变量 changeDirection 的值，该变量用于控制蛇的运动方向。在

本例中提供了两种控制蛇的方法，即用鼠标或者键盘上的 W、D、A、S 键让蛇向上、右、下、左移动。程序开始时，蛇会按照 changeDirection 预设的值向右移动，直到用户按下键盘改变其方向。

在程序开始的初始化部分有一个叫 direction 的变量，这个变量协同 changeDirection 检测用户发出的命令是否有效。蛇不应该立即向后运动（如果发生该情况，蛇会死亡，同时游戏结束）。为了防止这样的情况发生，将用户发出的请求（保存在 changeDirection 里）和目前的方向（保存在 direction 里）进行比较，如果方向相反，忽略该命令，蛇会继续按原方向运动。这里用下面几行代码进行比较：

```
if changeDirection=='right' and not direction=='left':
        direction=changeDirection
if changeDirection=='left' and not direction=='right':
        direction=changeDirection
if changeDirection=='up' and not  direction=='down':
        direction=changeDirection
if changeDirection=='down' and not direction=='up':
        direction=changeDirection
```

这样就保证了用户输入的合法性，蛇（在屏幕上显示为一系列块）就能够按照用户的输入移动。每次转弯时，蛇都会向该方向移动一小节。每个小节为 20 像素，用户可以告诉 Pygame 在任何方向移动一小节。

```
if direction=='right':
    snakePosition[0]+=20
if direction=='left':
    snakePosition[0]-=20
if direction=='up':
    snakePosition[1]-=20
if direction=='down':
    snakePosition[1]+=20
```

snakePosition 为蛇头的新位置，程序开始处的另一个列表变量 snakeSegments 却不是这样。该列表存储蛇身体的位置（头部后边），随着蛇吃掉草莓导致长度增加，列表会增加长度同时提高游戏难度。随着游戏的进行，避免蛇头撞到身体的难度变大。如果蛇头撞到身体，蛇会死亡，同时游戏结束。此时用下面的代码使蛇的身体增长：

```
snakeSegments.insert(0,list(snakePosition))
```

这里用 insert()方法向 snakeSegments 列表（存有蛇当前的位置）中添加新项目。每当 Python 运行到这一行，它就会将蛇的身体增加一节，同时将这一节放在蛇的头部，在玩家看来蛇在增长。当然，用户只希望蛇吃到草莓时才增长，否则蛇会一直变长。输入下面的几行代码：

```
if snakePosition[0]==raspberryPosition[0]
and snakePosition[1]==raspberryPosition[1]:
    raspberrySpawned=0
```

```
    else:
        snakeSegments.pop()
```

第 1 条 if 语句检查蛇头部的 x 和 y 坐标是否等于草莓（玩家的目标点）的坐标。如果等于，该草莓就会被蛇吃掉，同时 raspberrySpawned 变量置为 0。else 语句告诉 Python 如果草莓没有被吃掉，将 snakeSegments 列表中最早的项目 pop 出来。

pop 语句简单、易用，它返回列表中末尾的项目并从列表中删除，使列表缩短一项。在 snakeSegments 列表里，它使 Python 删掉距离头部最远的一部分。在玩家看来，蛇整体在移动而不会增长。实际上，它在一端增加小节，在另一端删除小节。由于有 else 语句，pop 语句只有在没吃到草莓时执行。如果吃到了草莓，列表中的最后一项不会被删掉，所以蛇会增加一小节。

现在，蛇就可以通过吃草莓让自己变长了。但是如果游戏中只有一个草莓会有些无聊，所以若蛇吃了一个草莓，用下面的代码增加一个新的草莓到游戏界面中：

```
    if raspberrySpawned==0:
        x=random.randrange(1,32)
        y=random.randrange(1,24)
        raspberryPosition=[int(x*20),int(y*20)]
    raspberrySpawned=1
```

这部分代码通过判断变量 raspberrySpawned 是否为 0 来判断草莓是否被吃掉了，如果被吃掉，使用程序开始引入的 random 模块获取一个随机的位置。然后将这个位置和蛇的每个小节的长度（20 像素宽，20 像素高）相乘来确定它在游戏界面中的位置。随机地放置草莓是很重要的，防止用户预先知道下一个草莓出现的位置。最后将 raspberrySpawned 变量置 1，以保证每个时刻界面上只有一个草莓。

现在有了让蛇移动和生长的代码，包括草莓被吃和新建的操作（在游戏中称为草莓重生），但是还没有在界面上画东西，输入下面的代码：

```
    playSurface.fill(blackColour)
    for position in snakeSegments:      #画蛇（一系列方块）
        pygame.draw.rect(playSurface,whiteColour,Rect(position[0],
         position[1], 20, 20))
    pygame.draw.rect(playSurface,redColour,Rect(raspberryPosition[0],
    raspberryPosition[1], 20, 20))      #草莓
    pygame.display.flip()
```

这些代码让 Pygame 填充背景色为黑色，蛇的头部和身体为白色，草莓为红色。最后一行的 pygame.display.flip()让 Pygame 更新界面（如果没有这条语句，用户将看不到任何东西。每次在界面上画完对象时，记得使用 pygame.display.flip()让用户看到更新）。

现在还没有涉及蛇死亡的代码。如果游戏中的角色永远死不了，玩家很快会感到无聊，所以用下面的代码设置一些让蛇死亡的场景：

```
    if snakePosition[0]>620 or snakePosition[0]<0:
        gameOver()
    if snakePosition[1]>460 or snakePosition[1]<0:
```

```
        gameOver()
```

第 1 个 if 语句检查蛇是否已经走出了界面的上、下边界，第 2 个 if 语句检查蛇是否已经走出了左、右边界。这两种情况都是蛇的末日，触发前面定义的 gameOver()函数，打印游戏结束信息并退出游戏。如果蛇头撞到了自己身体的任何部分，也会让蛇死亡，所以输入下面几行代码：

```
    for snakeBody in snakeSegments[1:]:
        if snakePosition[0]==snakeBody[0] and
        snakePosition[1]==snakeBody[1]:
            gameOver()
```

这里的 for 语句遍历蛇的每一小节的位置（从列表的第 2 项开始到最后一项），同时和当前蛇头的位置比较。这里用 snakeSegments[1:]来保证从列表的第 2 项开始遍历。列表的第 1 项为头部位置，如果从第 1 项开始比较，那么游戏一开始蛇就死亡了。

最后，只需要设置 fpsClock 变量的值即可控制游戏速度。

```
    fpsClock.tick(20)
```

使用 IDLE 的 Run Module 选项或者在终端中输入 python snake.py 来运行程序。

贪吃蛇 snake.py 的完整源代码如下：

```
    import pygame, sys, time, random
    from pygame.locals import *
    pygame.init()
    fpsClock=pygame.time.Clock()
    playSurface=pygame.display.set mode((640, 480))
    pygame.display.set caption('Raspberry Snake')
    #定义一些颜色
    redColour=pygame.Color(255, 0, 0)
    blackColour=pygame.Color(0, 0, 0)
    whiteColour=pygame.Color(255, 255, 255)
    greyColour=pygame.Color(150, 150, 150)
    #初始化程序中用到的一些变量
    snakePosition=[100,100]
    snakeSegments=[[100,100],[80,100],[60,100]]
    raspberryPosition=[300,300]     #草莓位置
    raspberrySpawned=1              #是否吃到草莓，1 为没有吃到，0 为吃到
    direction='right'               #运动方向
    changeDirection=direction
    def gameOver():
        gameOverFont=pygame.font.Font('simfang.ttf', 72)
        gameOverSurf=gameOverFont.render('Game Over', True, greyColour)
        gameOverRect=gameOverSurf.get rect()
        gameOverRect.midtop=(320, 10)
        playSurface.blit(gameOverSurf, gameOverRect)
        pygame.display.flip()
```

```
        time.sleep(5)
        pygame.quit()
        sys.exit()
    while True:
        for event in pygame.event.get():
            if event.type==QUIT:
                pygame.quit()
                sys.exit()
            elif event.type==KEYDOWN:
                if event.key==K_RIGHT or event.key==ord('d'):
                    changeDirection='right'
                if event.key==K_LEFT or event.key==ord('a'):
                    changeDirection='left'
                if event.key==K_UP or event.key==ord('w'):
                    changeDirection='up'
                if event.key==K_DOWN or event.key==ord('s'):
                    changeDirection='down'
                if event.key==K_ESCAPE:
                    pygame.event.post(pygame.event.Event(QUIT))
        if changeDirection=='right' and not direction=='left':
                direction=changeDirection
        if changeDirection=='left' and not direction=='right':
                direction=changeDirection
        if changeDirection=='up' and not  direction=='down':
                direction=changeDirection
        if changeDirection=='down' and not direction=='up':
                direction=changeDirection
        if direction=='right':
            snakePosition[0]+=20
        if direction=='left':
            snakePosition[0]-=20
        if direction=='up':
            snakePosition[1]-=20
        if direction=='down':
            snakePosition[1]+=20
        #将蛇的身体增加一节，同时将这一节放在蛇的头部
        snakeSegments.insert(0,list(snakePosition))
        #检查蛇头部的 x 和 y 坐标是否等于草莓（玩家的目标点）的坐标
        if snakePosition[0]==raspberryPosition[0] and snakePosition[1]==
        raspberryPosition[1]:
            raspberrySpawned=0
        else:
```

```
            snakeSegments.pop()
        #在游戏界面中增加一个新的草莓
        if raspberrySpawned==0:
            x=random.randrange(1,32)
            y=random.randrange(1,24)
            raspberryPosition = [int(x*20),int(y*20)]
        raspberrySpawned=1
        playSurface.fill(blackColour)
        for position in snakeSegments:      #画蛇（一系列方块）
            pygame.draw.rect(playSurface,whiteColour,Rect(position[0],
            position[1], 20, 20))
        #画草莓
        pygame.draw.rect(playSurface,redColour,Rect(raspberryPosition[0],
        raspberryPosition[1], 20, 20))
        pygame.display.flip()
        if snakePosition[0]>620 or snakePosition[0] < 0:
            gameOver()
        if snakePosition[1]>460 or snakePosition[1] < 0:
            gameOver()
        for snakeBody in snakeSegments[1:]:
            if snakePosition[0]==snakeBody[0] and snakePosition[1]==
            snakeBody[1]:
              gameOver()
        fpsClock.tick(10)
```

17.4　基于 Pygame 设计飞机大战游戏

相信玩过《雷电》的朋友都熟悉打飞机，这里将游戏做了简化。飞机的速度固定，子弹的速度固定，基本操作是通过键盘移动玩家飞机，敌机随机从屏幕上方出现并匀速落到下方，子弹从玩家飞机发出，碰到目标飞机会击毁，如果目标飞机碰到玩家飞机，则游戏结束并显示分数。飞机大战游戏的运行效果如图 17-8 所示。

17.4.1　游戏角色

本游戏中所需的角色包括玩家飞机、敌机及子弹。用户可以通过键盘移动玩家飞机在屏幕上的位置来打不同位置的敌机，因此设计玩家类 Player、敌机类 Enemy 和子弹类 Bullet 对应 3 种游戏角色。

对于玩家类 Player，需要的操作有射击和移动两种，移动又分为上、下、左、右 4 种情况。

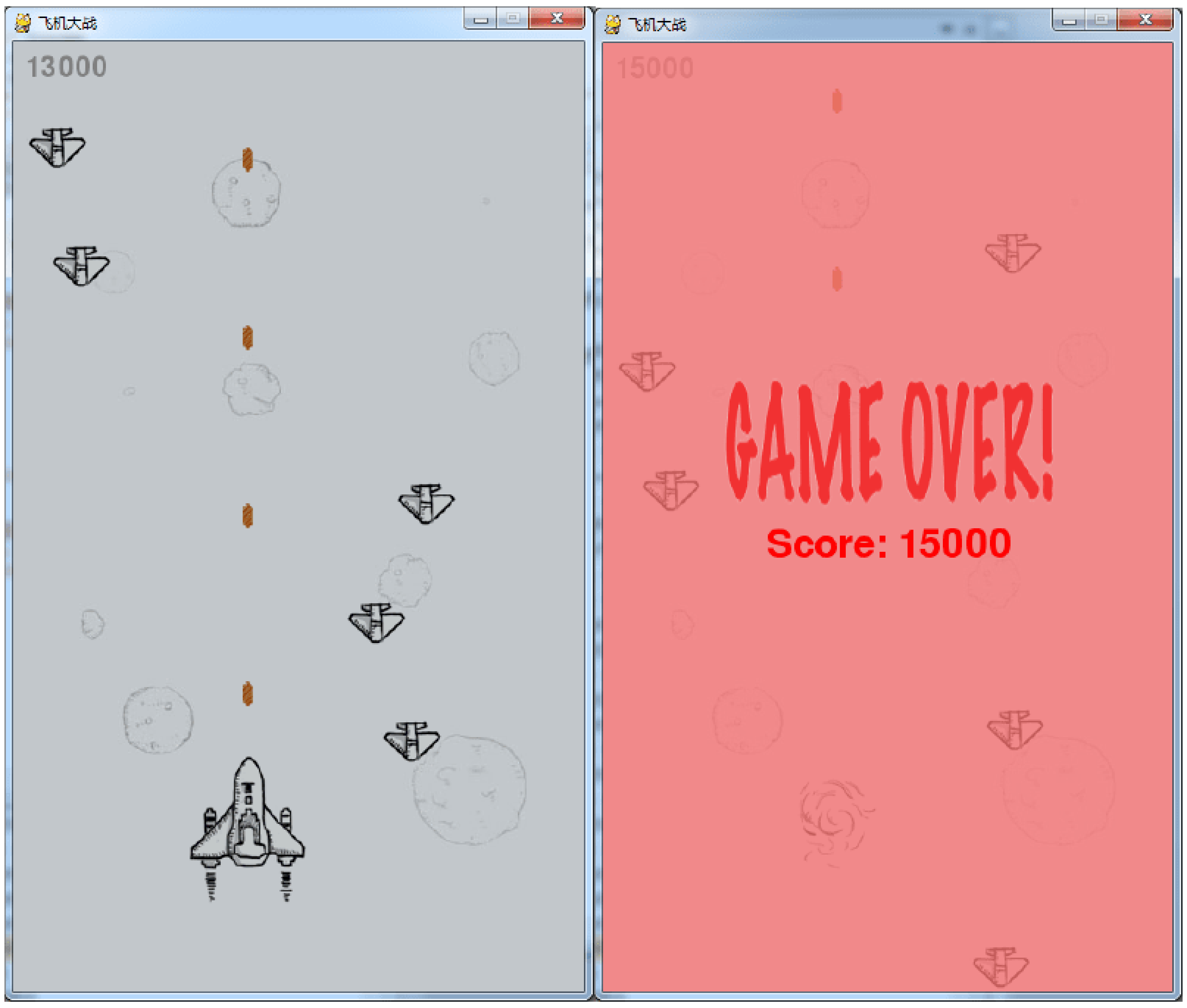


图 17-8　飞机大战游戏的运行效果

对于敌机类 Enemy，则比较简单，只需要移动即可，从屏幕上方出现并移动到屏幕下方。

对于子弹类 Bullet，与飞机相同，仅需要以一定的速度移动即可。

玩家、子弹和敌机都可以写成一个类，继承 Pygame 的 Sprite 类，实现一些动画效果以及检测碰撞。

```
import pygame
from sys import exit
from pygame.locals import *
#from gameRole import *
import random

SCREEN_WIDTH=480
SCREEN_HEIGHT=800
TYPE_SMALL=1
TYPE_MIDDLE=2
TYPE_BIG=3

#子弹类
class Bullet(pygame.sprite.Sprite):                    #继承 Sprite 精灵类
    def __init__(self, bullet_img, init_pos):
        pygame.sprite.Sprite.__init__(self)
        self.image=bullet_img
        self.rect=self.image.get_rect()
        self.rect.midbottom=init_pos
```

```
        self.speed=10
    def move(self):
        self.rect.top-=self.speed
#玩家类
class Player(pygame.sprite.Sprite):            #继承 Sprite 精灵类
    def __init__(self, plane img, player rect, init pos):
        pygame.sprite.Sprite.__init__(self)
        self.image=[]                          #用来存储玩家对象精灵图片的列表
        for i in range(len(player rect)):
            self.image.append(plane_img.subsurface(player_rect[i])
            .convert alpha())
        self.rect=player_rect[0]           #初始化图片所在的矩形
        self.rect.topleft=init pos         #初始化矩形的左上角坐标
        self.speed=8                       #初始化玩家速度，这里是一个确定的值
        self.bullets=pygame.sprite.Group() #玩家飞机所发射的子弹的集合
        self.img_index=0                   #玩家精灵图片索引
        self.is_hit=False                  #玩家是否被击中
    def shoot(self, bullet img):
        bullet=Bullet(bullet img, self.rect.midtop)
        self.bullets.add(bullet)
    def moveUp(self):
        if self.rect.top <= 0:
            self.rect.top=0
        else:
            self.rect.top-=self.speed
    def moveDown(self):
        if self.rect.top>=SCREEN HEIGHT-self.rect.height:
            self.rect.top=SCREEN HEIGHT-self.rect.height
        else:
            self.rect.top+=self.speed
    def moveLeft(self):
        if self.rect.left<=0:
            self.rect.left=0
        else:
            self.rect.left-=self.speed
    def moveRight(self):
        if self.rect.left>=SCREEN WIDTH-self.rect.width:
            self.rect.left=SCREEN WIDTH-self.rect.width
        else:
            self.rect.left+=self.speed
#敌机类
class Enemy(pygame.sprite.Sprite):         #继承 Sprite 精灵类
```

```
    def __init__(self, enemy_img, enemy_down_imgs, init_pos):
      pygame.sprite.Sprite.__init__(self)
      self.image=enemy_img
      self.rect=self.image.get_rect()
      self.rect.topleft=init_pos
      self.down_imgs=enemy_down_imgs
      self.speed=2
      self.down_index=0
    def move(self):
       self.rect.top+=self.speed
```

以上设计了游戏中的 3 个角色。

17.4.2　游戏界面显示

在游戏画面中使用了一些飞机、子弹图像，这里使用 shoot.png 文件（如图 17-9 所示）存储所有飞机、子弹、爆炸等图像，在程序中需要分割出来显示。当然，可以用图像处理软件分解成一个个独立文件，这样处理后开发程序简单些。

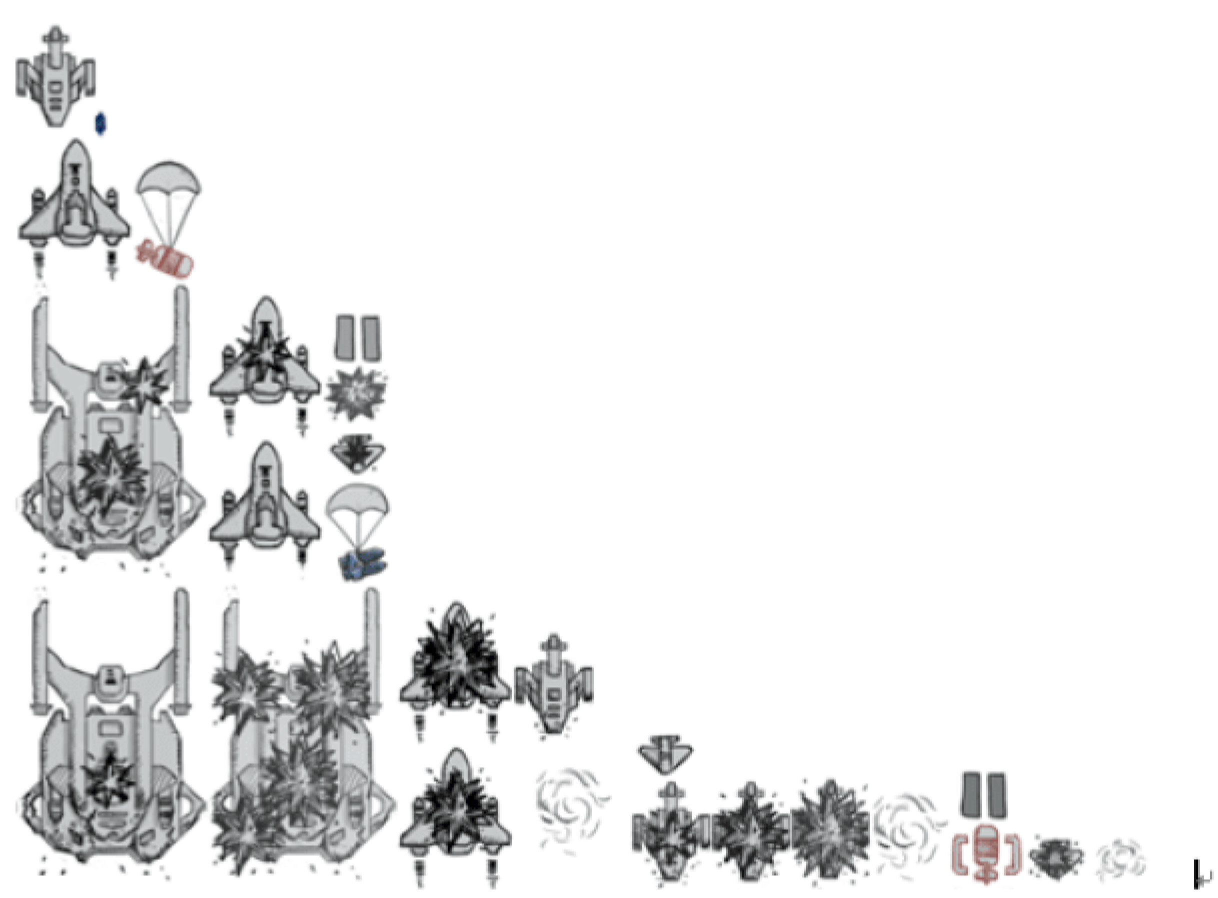


图 17-9　飞机大战游戏的图像文件 shoot.png

所有的飞机都在 shoot.png 图片中。在游戏中显示的元素（包括飞机、子弹等）在 Pygame 中都是一个 surface，这时可以利用 Pygame 提供的 subsurface()方法，首先载入一张大图，然后调用 subsurface()方法选取其中的一小部分生成一个新的 surface。

```
#载入飞机图片
plane_img=pygame.image.load('resources/image/shoot.png')
#选择飞机在大图中的位置，并生成 subsurface，然后初始化飞机开始的位置
player_rect=pygame.Rect(0, 99, 102, 126)
player1=plane_img.subsurface(player_rect)      #获取飞机图片
player_pos=[200, 600]
screen.blit(player1, player_pos)               #绘制飞机
```

初始化游戏并根据设置好的大小生成游戏窗口；载入游戏音乐、背景图片 background.png、游戏结束画面 gameover.png 以及飞机和子弹图像 shoot.png；设置相关参数。最后定义存储敌人的飞机精灵组 enemies1 和用来渲染击毁精灵动画的爆炸飞机精灵组 enemies_down。

```
#初始化游戏
pygame.init()
screen=pygame.display.set mode((SCREEN WIDTH, SCREEN HEIGHT))
pygame.display.set caption('飞机大战')

#载入游戏音乐
bullet sound=pygame.mixer.Sound('resources/sound/bullet.wav')
enemy1 down sound=pygame.mixer.Sound('resources/sound/enemy1 down.wav')
game over sound=pygame.mixer.Sound('resources/sound/game over.wav')
bullet sound.set volume(0.3)
enemy1 down sound.set volume(0.3)
game over sound.set volume(0.3)
pygame.mixer.music.load('resources/sound/game music.wav')
pygame.mixer.music.play(-1, 0.0)
pygame.mixer.music.set volume(0.25)

background=pygame.image.load('resources/image/background.png')
.convert()                                  #载入背景图
game over=pygame.image.load('resources/image/gameover.png')
                                            #载入游戏结束图 gameover.png
filename='resources/image/shoot.png'
plane_img=pygame.image.load(filename)       #载入飞机和子弹图 shoot.png

#设置玩家相关参数
player rect=[]
player rect.append(pygame.Rect(0, 99, 102, 126))       #玩家精灵图片区域
player rect.append(pygame.Rect(165, 360, 102, 126))
player_rect.append(pygame.Rect(165, 234, 102, 126))    #玩家爆炸精灵图片区域
player rect.append(pygame.Rect(330, 624, 102, 126))
player rect.append(pygame.Rect(330, 498, 102, 126))
player rect.append(pygame.Rect(432, 624, 102, 126))
player pos=[200, 600]
player=Player(plane_img, player_rect, player_pos)

#定义子弹对象使用的 surface 相关参数
bullet rect=pygame.Rect(1004, 987, 9, 21)
bullet img=plane img.subsurface(bullet rect)

#定义敌机对象使用的 surface 相关参数
enemy1 rect=pygame.Rect(534, 612, 57, 43)
enemy1_img=plane_img.subsurface(enemy1_rect)
```

```
enemy1_down_imgs=[]
enemy1_down_imgs.append(plane_img.subsurface(pygame.Rect(267, 347, 57,
43)))
enemy1_down_imgs.append(plane_img.subsurface(pygame.Rect(873, 697, 57,
43)))
enemy1_down_imgs.append(plane_img.subsurface(pygame.Rect(267, 296, 57,
43)))
enemy1_down_imgs.append(plane_img.subsurface(pygame.Rect(930, 697, 57,
43)))
enemies1=pygame.sprite.Group()          #存储敌人的飞机
enemies_down=pygame.sprite.Group()      #存储被击毁的飞机，用来渲染击毁精灵动画
shoot_frequency=0
enemy_frequency=0
player_down_index=16
score=0
clock=pygame.time.Clock()
running=True
```

17.4.3　游戏的逻辑实现

下面进入游戏主循环，在主循环中进行了以下操作：

❶ 处理键盘输入的事件，增加游戏操作交互

处理键盘输入的事件指上、下、左、右按键操作，增加游戏操作交互指玩家飞机的上、下、左、右移动。

```
key_pressed=pygame.key.get_pressed()
#若玩家被击中，则无效
if not player.is_hit:
    if key_pressed[K_w] or key_pressed[K_UP]:   #处理键盘事件（移动飞机的位置）
        player.moveUp()
    if key_pressed[K_s] or key_pressed[K_DOWN]:#处理键盘事件（移动飞机的位置）
        player.moveDown()
    if key_pressed[K_a] or key_pressed[K_LEFT]:#处理键盘事件（移动飞机的位置）
        player.moveLeft()
    if key_pressed[K_d] or key_pressed[K_RIGHT]: #处理键盘事件（移动飞机的位置）
        player.moveRight()
```

❷ 处理子弹

这里控制发射子弹的频率并发射子弹，移动已发射过的子弹，若超出窗口范围则删除。

```
#控制发射子弹的频率并发射子弹
if not player.is_hit:   #（1）判断玩家飞机没有被击中
    if shoot_frequency%15==0:
```

```
        bullet sound.play()
        player.shoot(bullet img)
    shoot frequency+=1
    if shoot frequency>=15:
        shoot_frequency=0
#移动已发射过的子弹，若超出窗口范围则删除
for bullet in player.bullets:
    bullet.move()                          #(2)以固定速度移动子弹
    if bullet.rect.bottom<0:               #(3)子弹移动出屏幕后删除子弹
        player.bullets.remove(bullet)      #删除子弹
```

❸ 敌机处理

敌机需要在界面上方随机产生，并以一定的速度向下移动，详细步骤如下：

（1）生成敌机，需要控制生成频率。

（2）移动敌机。

（3）敌机与玩家飞机碰撞效果的处理。

（4）移出屏幕后删除敌机。

（5）敌机被子弹击中效果的处理。

❹ 得分显示

在游戏界面的固定位置显示消灭了多少目标敌机。

```
score font=pygame.font.Font(None, 36)
score text=score font.render(str(score), True, (128, 128, 128))text rect
=score text.get rect()
text rect.topleft=[10, 10]
screen.blit(score_text, text_rect)
```

游戏主循环的完整代码如下：

```
while running:
    clock.tick(60)                         #控制游戏最大帧率为 60
    #控制发射子弹的频率并发射子弹
    if not player.is hit:
        if shoot frequency % 15==0:
            bullet sound.play()
            player.shoot(bullet img)
        shoot frequency+=1
        if shoot frequency>=15:
            shoot frequency=0
    #移动子弹，若超出窗口范围则删除
    for bullet in player.bullets:
        bullet.move()
        if bullet.rect.bottom<0:
            player.bullets.remove(bullet)

    #生成敌机
```

```
if enemy_frequency % 50==0:              #(1)生成敌机，需要控制生成频率
    enemy1_pos=[random.randint(0, SCREEN_WIDTH - enemy1_rect.width), 0]
    enemy1=Enemy(enemy1_img, enemy1_down_imgs, enemy1_pos)
    enemies1.add(enemy1)
enemy_frequency+=1
if enemy_frequency>=100:
    enemy_frequency=0

#移动敌机，若超出窗口范围则删除
for enemy in enemies1:
    enemy.move()                         #(2)移动敌机
    #判断玩家是否被击中
    if pygame.sprite.collide_circle(enemy, player):
                                         #(3)敌机与玩家飞机碰撞效果的处理
        enemies_down.add(enemy)
        enemies1.remove(enemy)
        player.is_hit=True
        game_over_sound.play()
        break
    if enemy.rect.top > SCREEN_HEIGHT:   #(4)移出屏幕后删除飞机
        enemies1.remove(enemy)
#(5)敌机被子弹击中效果的处理
#将被击中的敌机对象添加到击毁敌机 Group 中，用来渲染击毁动画
enemies1_down=pygame.sprite.groupcollide(enemies1,player.bullets,1,1)
for enemy_down in enemies1_down:
    enemies_down.add(enemy_down)

#绘制背景
screen.fill(0)
screen.blit(background, (0, 0))

#绘制玩家飞机
if not player.is_hit:
    screen.blit(player.image[player.img_index], player.rect)
    #更换图片索引使飞机有动画效果
    player.img_index=shoot_frequency       // 8
else:
    player.img_index=player_down_index     // 8
    screen.blit(player.image[player.img_index], player.rect)
    player_down_index+=1
    if player_down_index>47:
        running=False
#绘制击毁动画
for enemy_down in enemies_down:
```

```
        if enemy down.down index==0:
            enemy1 down sound.play()
        if enemy down.down index>7:
            enemies down.remove(enemy down)
            score+=1000
            continue
        screen.blit(enemy down.down imgs[enemy down.down index // 2],
        enemy down.rect)
        enemy down.down index+=1
    #绘制子弹和敌机
    player.bullets.draw(screen)
    enemies1.draw(screen)
    #绘制得分
    score font=pygame.font.Font(None, 36)
    score text=score font.render(str(score), True, (128, 128, 128))
    text rect=score text.get rect()
    text rect.topleft=[10, 10]
    screen.blit(score text, text rect)
    #更新屏幕
    pygame.display.update()
    for event in pygame.event.get():
        if event.type==pygame.QUIT:
            pygame.quit()
            exit()
    #监听键盘事件
    key pressed=pygame.key.get pressed()
    #若玩家被击中，则无效
    if not player.is hit:
        if key_pressed[K_w] or key_pressed[K_UP]:
            player.moveUp()
        if key pressed[K s] or key pressed[K DOWN]:
            player.moveDown()
        if key pressed[K a] or key pressed[K LEFT]:
            player.moveLeft()
        if key pressed[K d] or key pressed[K RIGHT]:
            player.moveRight()
font=pygame.font.Font(None, 48)
text=font.render('Score: '+ str(score), True, (255, 0, 0))
text_rect=text.get_rect()
text rect.centerx=screen.get rect().centerx
text rect.centery=screen.get rect().centery+24
screen.blit(game over, (0, 0))
screen.blit(text, text rect)
while 1:
    for event in pygame.event.get():
```

```
            if event.type==pygame.QUIT:
                pygame.quit()
                exit()
        pygame.display.update()
```

目前基本实现了玩家移动并发射子弹、随机生成敌机、击中敌机并爆炸、玩家飞机被击毁、背景音乐及音效、游戏结束并显示分数这几项功能，已经是一个简单、可玩的游戏。整个游戏的实现不到 300 行代码，可以看出 Python 代码是多么简洁和高效。

第18章

机器学习案例——基于朴素贝叶斯算法的文本分类

18.1 文本分类功能介绍

视频讲解

对于分类问题，其实大家都很熟悉，说每个人每天都在执行分类操作一点也不夸张，只是大家没有意识到罢了。例如，当看到一个陌生人，脑子里会下意识地判断是男是女；走在路上，可能经常会对身旁的朋友说“这个人一看就很有钱”“这个人看着很谦和”之类的话，其实这就是一种分类操作，也就是根据这个人身上的某些特征做出的一个推断。

文本分类和其他分类本质上是一样的，只不过分类的对象是文本。文本分类问题是根据文本的特征将其分到预先设定好的类别中，类别可以是两类，也可以是更多的类别。文本数据是互联网时代的一种最常见的数据形式，新闻报道、电子邮件、网页、评论留言、博客文章、学术论文等都是常见的文本数据类型，文本分类问题所采用的类别划分往往也会根据目的不同而有较大差别。例如，根据文本内容可以有“政治”“经济”“体育”等不同类别；根据应用目的要求，在检测垃圾邮件时可以有“垃圾邮件”和“非垃圾邮件”；根据文本特点，在做情感分析时可以有“积极情感文本”和“消极情感文本”。可见，文本分类的应用非常广泛。

常用的文本分类算法有朴素贝叶斯算法、支持向量机算法、决策树、KNN 最近邻算法等。KNN 最近邻算法的原理最简单，但是分类精度一般，速度很慢；朴素贝叶斯算法对于短文本分类效果最好，精度很高；支持向量机算法的优势是支持线性不可分的情况，在精度上取中。本章以朴素贝叶斯算法为例讲解文本分类的一般过程，并以垃圾邮件过滤作为朴素贝叶斯算法的一个应用案例进行测试。

18.2 程序设计的思路

文本分类主要有 3 个步骤：

❶ **文本的表达**

这个阶段的主要任务是将文档转变成矢量形式，常用且最简单的就是词集模型。其基本思想是假定对于一个文档忽略其词序、语法和句法等要素，将文档仅仅看作是若干词汇的集合，而文档中每个单词的出现都是独立的，不依赖于其他词汇的出现。简单来说，就是将每篇文档都看成一个词汇集合，然后看这个集合里出现了什么词汇，将其分类。例如有以下两个文档。

文档一：Bob likes to play basketball,Jim likes too.

文档二：Bob also likes to play football games.

基于这两个文档，先构造一个词汇表：

```
wordList={"Bob","likes","to","play","basketball","Jim","too","also",
"football","games"}
```

可以看到，这个词汇表由 10 个不同的单词组成，利用单词在词汇表中的索引，上面两个文档都可以用一个 10 维向量表示，向量中的每个元素表示词汇表中的相应单词在文档中是否出现，出现为 1，不出现为 0。

文档一：[1,1,1,1,1,1,1,0,0,0]

文档二：[1,1,1,1,0,0,0,1,1,1]

❷ **分类器的选择与训练**

这个阶段是文本分类的关键步骤，使用的分类算法不同，具体步骤也不同，这里使用朴素贝叶斯算法进行文本分类，具体将在 18.3 节详细介绍。

❸ **分类结果的评价**

机器学习领域的算法评估常用的评价指标如下：

错分率=所有分类错误的记录数/测试集中的记录总数

准确率=被识别为该分类的正确分类记录数/被识别为该分类的记录数

召回率=被识别为该分类的正确分类记录数/测试集中该分类的记录总数

F1-score=2(准确率×召回率)/(准确率+召回率)

18.3 关键技术

18.3.1 贝叶斯算法的理论基础

❶ **条件概率**

假设一个盒子里装了 9 个小球，如图 18-1 所示，其中 4 个黑色的、5 个白色的，如果从盒子里随机取一个小球，那么是黑色小球的可能性有多大？由于取小球有 9 种可能，其中 4 种为黑色，所以取出黑色小球的概率为 4/9。那么取出白色小球的概率又会是多少呢？很显然，是 5/9。如果用 P(black)表示取黑色小球的概率，其概率值等于黑色小球的数量除以小球总数。

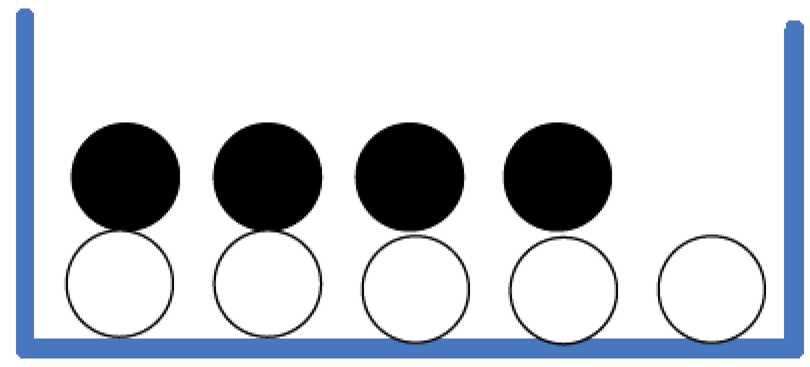


图 18-1　一个装有 9 个小球的盒子

如果将 9 个小球分开放在两个盒子中，如图 18-2 所示，那么上述概率应该如何计算？



A 盒　　　　　　　　B 盒

图 18-2　将 9 个小球分别放在两个盒子中

将 9 个小球如图 18-2 所示放在两个盒子中，要计算 P(black)或者 P(white)，事先得知小球所在盒子的信息会不会改变结果？假设从 A 盒中取出黑色小球的概率为 P(black|A 盒)，即“在已知小球取自 A 盒的情况下取出黑色小球的概率”，容易得到从 A 盒中取出黑色小球的概率 P(black|A 盒)为 1/4，这就是所谓的条件概率。

条件概率的定义：已知事件 B 发生的条件下事件 A 发生的概率称为事件 A 关于事件 B 的条件概率，记为 P(A|B)。对于任意事件 A 和 B，若 P(B)≠0，则“在事件 B 发生的条件下事件 A 发生的条件概率”记为 P(A|B)，定义为$P(A\,|\,B)=\dfrac{P(AB)}{P(B)}$。

那么，对于小球这个例子来说，依据条件概率的定义，P(black|A 盒)=P(black and A 盒)/P(A 盒)。

下面来看上述公式是否合理。P(black and A 盒)表示取出 A 盒中黑色小球的概率，用 A 盒中的黑色小球数量除以小球总数可得，即 1/9；P(A 盒)表示从 A 盒中取小球的可能性，即用 A 盒中的小球数量除以小球总数，即 4/9。于是有 P(black|A 盒)=P(black and A 盒)/P(A 盒)=(1/9)/(4/9)=1/4，结论与前面分析得到的结果相同，说明公式完全合理。

❷ 全概率公式

若事件组$(A_1,A_2,\cdots,A_n)$满足以下关系：

（1）$A_i(i=1,2,\cdots,n)$两两互斥，且$P(A_i)>0$。

（2）$\sum_1^n A_i=\Omega$，Ω为样本空间。

则称事件组$(A_1,A_2,\cdots,A_n)$是样本空间Ω的一个划分。

全概率公式：设$(A_1,A_2,\cdots,A_n)$是样本空间Ω的一个划分，B为任一事件，则有

$$P(B)=\sum_{i=1}^{n}P(A_i)P(B|A_i)$$

❸ 贝叶斯公式

设$(A_1,A_2,\cdots,A_n)$是样本空间Ω的一个划分，B为任一事件，则有：

$$P(A_i \mid B)=\frac{P(A_i B)}{P(B)}=\frac{P(A_i)P(B|A_i)}{\sum_{j=1}^{n}P(A_j)P(B \mid A_j)}$$

上式中的 A_i 常被视为导致试验结果 B 发生的“原因”，$P(A_i)$（$i=1,2,\cdots,n$）表示各种原因发生的可能性大小，故称先验概率；$P(A_i|B)$（$i=1,2,\cdots,n$）则反映当试验产生了结果 B 之后再对各种原因概率的新认识，故称后验概率。

18.3.2　朴素贝叶斯分类

❶ 朴素贝叶斯分类的原理

朴素贝叶斯分类基于条件概率、贝叶斯公式和独立性假设原则。

基于概率论的方法告诉我们，当只有两种分类时：

如果 P1(x,y)>P2(x,y)，那么分入类别 1；

如果 P2(x,y)>P1(x,y)，那么分入类别 2。

这里提到的 P1(x,y)、P2(x,y)是一种简化描述，分别表示 $P(c_1|x,y)$和 $P(c_2|x,y)$，这些符号表示的意义可以这样理解，给定某个由 x、y 两个特征表示的数据点，那么该数据点是类别 c_1 的概率是多少？是类别 c_2 的概率又是多少？应用贝叶斯定理可得：

$$P(c_i \mid x,y)=\frac{P(x,y \mid c_i)P(c_i)}{p(x,y)}$$

从而可以定义贝叶斯分类准则如下：

- 如果 $p(c_1|x,y)> p(c_2|x,y)$，那么属于类别 c_1；
- 如果 $p(c_1|x,y)<p(c_2|x,y)$，那么属于类别 c_2。

这样，使用贝叶斯公式可以通过计算已知的 3 个概率值来得到未知的概率值。如果仅仅为了比较 $P(c_1|x, y)$和 $P(c_2|x, y)$的大小，因为分母相同，只需要已知分子上的两个概率即可。这里还有一个假设，就是基于特征条件独立的假设，也就是上面说到的数据点的两个特征 x 和 y 相互独立，不会相互影响，因此可以将 $P(x,y|c_i)$展开成独立事件概率相乘的形式，则有：

$$P(x,y|c_i)=P(x|c_i)P(y|c_i)$$

当然，对于多个特征条件的情况，上式仍然成立，这样计算概率就简单多了。

❷ 朴素贝叶斯分类的定义

扩展到一般情况，朴素贝叶斯分类的正式定义如下：

（1）设 $x=\{a_1, a_2, \cdots, a_m\}$为一个待分类项，$a_i$ 为 x 的一个特征属性，有类别 $C=\{y_1, y_2, \cdots, y_n\}$。

（2）计算 $p(y_1|x)$，$p(y_2|x)$，…，$p(y_n|x)$。

（3）如果 $p(y_k|x)=\max\{p(y_1|x), p(y_2|x),\cdots, p(y_n|x)\}$，则 $x\in y_k$。

现在的关键是如何计算第（2）步中的各个条件概率，可以这么做：

① 找到一个已知分类的待分类项集合，这个集合称为训练样本集。

② 统计得到各类别下各个特征属性的条件概率估计，即 $P(a_1|y_1)$，$P(a_2|y_1)$，…，$P(a_m|y_1)$；$P(a_1|y_2)$，$P(a_2|y_2)$，…，$P(a_m|y_2)$；…；$P(a_1|y_n)$，$P(a_2|y_n)$，…，$P(a_m|y_n)$。

③ 如果各个特征属性是条件独立的，则根据贝叶斯公式有如下推导：

$$P(y_i \mid x) = \frac{P(x \mid y_i)P(y_i)}{p(x)}$$

因为分母对于所有类别相同，所以只要将分子最大化即可。又因为各特征属性是条件独立的，所以有：

$$P(x \mid y_i)P(y_i) = P(a_1 \mid y_i)P(a_2|y_i)4\ \ P(a_m \mid y_i)P(y_i) = P(y_i)\prod_{j=1}^{m} P(a_j \mid y_i)$$

❸ 朴素贝叶斯分类的流程

根据上述分析，朴素贝叶斯分类的流程可以用图 18-3 表示。

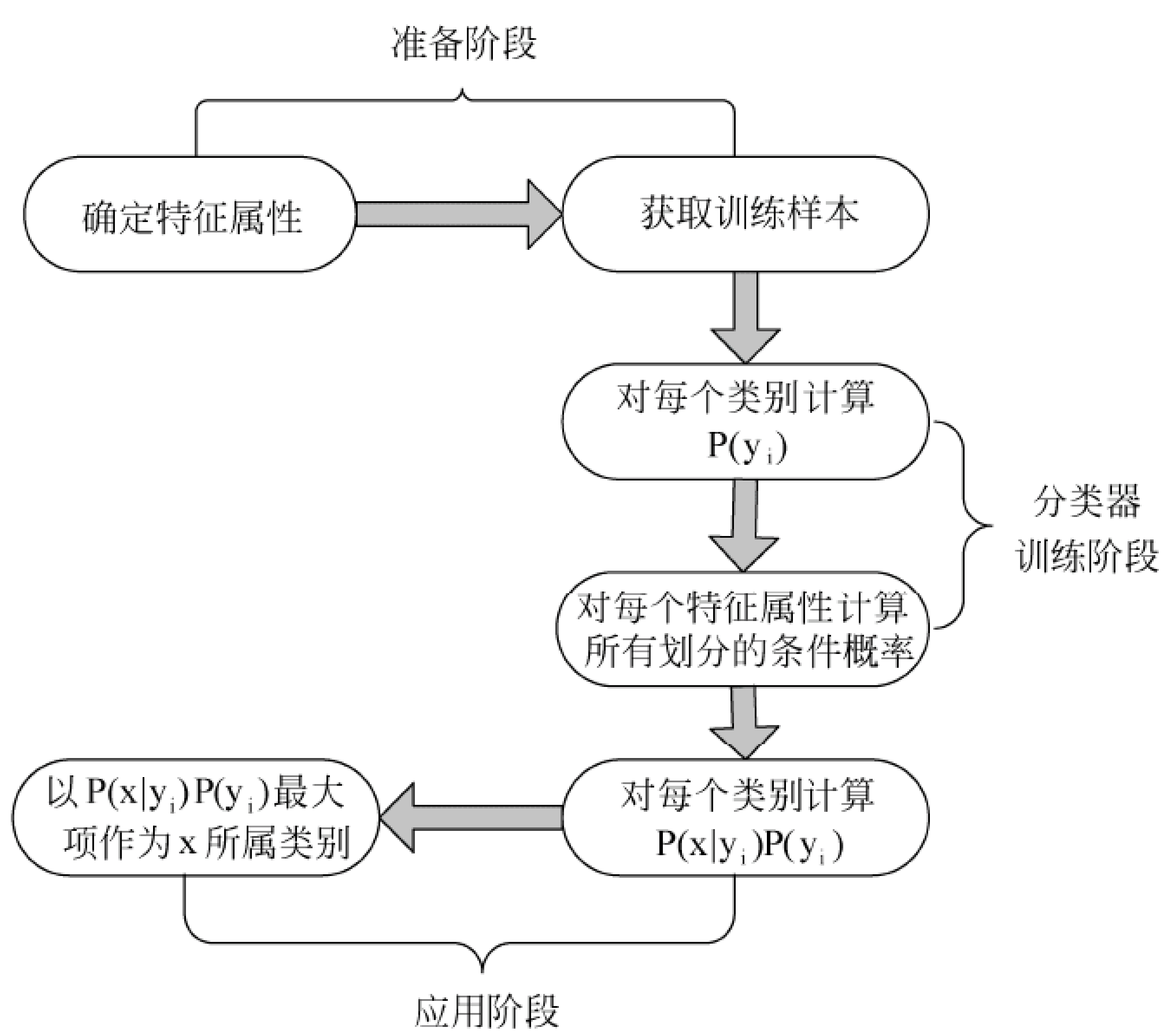


图 18-3　朴素贝叶斯分类流程

可以看到，整个朴素贝叶斯分类分为 3 个阶段。

（1）准备阶段：这个阶段为朴素贝叶斯分类做必要的准备，主要工作是根据具体应用情况确定特征属性，并对每个特征属性进行适当划分，然后由人工对一部分待分类项进行分类，形成训练样本集合。这一阶段的输入是所有待分类数据，输出是特征属性和训练样本集合。这一阶段是整个朴素贝叶斯分类中非常重要的一个阶段，也是唯一需要人工完成的阶段，其质量对整个过程有重要影响，分类器的质量在很大程度上由特征属性、特征属性划分及训练样本质量决定。

（2）分类器训练阶段：这个阶段就是生成分类器，主要工作是计算每个类别在训练样本中的出现概率及每个特征属性划分对每个类别的条件概率估计，并将结果记录。其输入

是特征属性和训练样本，输出是分类器。这一阶段是自动化阶段，根据前面讨论的公式可以由程序自动计算完成。

（3）应用阶段：这个阶段的任务是使用分类器对待分类项进行分类，其输入是分类器和待分类项，输出是待分类项与类别的映射关系。这一阶段也是自动化阶段，由程序完成。

18.3.3　使用 Python 进行文本分类

在文本分类中，要从文本中获取特征，需要先拆分文本，以一封电子邮件为例，电子邮件中的某些元素构成特征。我们可以解析得到文本中的词，并把每个词作为一个特征，而每个词是否出现作为该特征的值，然后将每一个文本表示为一个词条向量，就是前面提到的词集模型。每一个文本的词条向量的大小与词汇表中词的数目一致。

假设特征之间相互独立。所谓独立指的是统计意义上的独立，即一个特征或者单词出现的可能性与它和其他单词相邻没有关系，比如说，“今天天气真好！”中的“今天”和“天气”出现的概率与这两个词相邻没有任何关系。这个假设正是朴素贝叶斯分类器中“朴素”一词的含义。朴素贝叶斯分类器中的另一个假设是每个特征同等重要。

18.4　程序设计的步骤

掌握了上面的基本理论，就可以开启文本分类之旅了，本节以判断留言板的留言是否为侮辱性言论为例，详细讲解使用朴素贝叶斯算法进行文本分类的过程。

问题概述：构建一个快速过滤器来屏蔽在线社区留言板上的侮辱性言论。如果某条留言中出现了负面或者侮辱性的词汇，就将该留言标识为内容不当。对此问题建立两个类别——侮辱类和非侮辱类，分别用 1 和 0 表示。

现在正式开始，首先创建一个名为 Nbayes.py 的新文件，然后依次将程序清单添加到文件中。

18.4.1　收集训练数据

为了将主要精力集中在分类算法本身，直接用简单的英文语料作为数据集。在实际应用中，可以通过爬虫或其他途径获取真实数据。

视频讲解

```
def loadDataSet():
    postingList=[['my','dog','has','flea','problems','help','please'],
                 ['maybe','not','take','him','to','dog','park','stupid'],
                 ['my','dalmation','is','so','cute','I','love','him'],
                 ['stop','posting','stupid','worthless','garbage'],
                 ['mr','licks','ate','my','steak','how','to','stop','him'],
                 ['quit','buying','worthless','dog','food','stupid']]
```

```
    classVec=[0,1,0,1,0,1]                   #1 代表侮辱性言论，0 代表正常言论
    return postingList,classVec
```

loadDataSet()函数直接用模拟的 6 个已分词的小文档和对应的 6 个类别标签作为训练数据。该函数返回两个列表，其中 postingList 表示已分词的文档列表，classVec 表示文档对应的类别列表。

18.4.2　准备数据

有了数据集，接下来需要从数据集生成文本的结构化描述方法，即向量空间模型，把文本表示为一个向量，把向量的每个特征表示为文本中出现的词，这里用前面提到的词集模型，首先遍历训练集中的所有文本，生成词汇表。

```
def vocabList(dataSet):
    vocabSet=set([])                         #使用 set 创建不重复的词汇集
    for document in dataSet:
        vocabSet=vocabSet|set(document)      #创建两个集合的并集
    return list(vocabSet)
```

vocabList(dataSet)函数的参数 dataSet 为文档集的单词列表，即 loadDataSet()函数的 postingList 返回值。该函数根据文档数据集 dataSet 创建一个包含在所有文档中不重复出现的单词列表，为此使用 Python 的 set 数据类型。首先创建一个空的集合，然后将每个文档的词集合加入该集合中，操作符“|”表示求两个集合的并集。

下面根据词汇表生成每篇文档的词集模型表示，即词向量。

```
def setOfWordsVec(vocabList,inputText):
    textVec=[0]*len(vocabList)             #创建一个所有元素都为 0 的向量
    #遍历文档中的所有单词，若出现了词汇表中的单词，则令文档向量中的对应值为 1
    for word in inputText:
      if word in vocabList:
          textVec[vocabList.index(word)]=1
    return textVec
```

setOfWordsVec(vocabList,inputText)函数有两个参数，vocabList 表示已知的词汇表，inputText 表示某一文档的单词列表，函数返回值 textVec 为文档 inputText 的词向量，向量中的每一元素为 0 或 1，表示词汇表中的元素在文档中是否出现，没有出现为 0，出现为 1。向量的长度与词汇表的长度相同。该函数首先创建一个和词汇表一样长的向量，向量元素全部为 0，接着遍历文档中的每个单词，判断该单词是否在词汇表中，如果是，就将文档词向量相应位置的元素置为 1。

18.4.3　分析数据

现在可以检查函数的执行情况，首先检查单词列表，看有无遗漏或重复单词，确保数据解析的正确性。保存 Nbayes.py 文件，运行该文件，然后在 Python 提示符下输入：

```
>>> listOposts,listClasses=loadDataSet()
>>> wordList=vocabList(listOposts)
>>> wordList
['dog', 'stop', 'my', 'has', 'worthless', 'licks', 'mr', 'ate', 'I', 'steak',
'please', 'maybe', 'help', 'park', 'food', 'quit', 'is', 'so', 'buying',
'love', 'dalmation', 'take', 'him', 'posting', 'how', 'stupid', 'not',
'flea', 'problems', 'garbage', 'cute', 'to']
```

经检查没有遗漏单词，也没有重复单词，这样就生成了共有 32 个单词的词汇表。

下面检查 setOfWordsVec()函数的执行效果，生成第 1 篇文档的词向量：

```
>>> setOfWordsVec(wordList,listOposts[0])
[1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 1, 0, 0, 0]
```

可以看到文档向量中索引为 0 的元素为 1，对应词汇表中的 dog，可以查看第 1 篇文档 ['my','dog','has','flea','problems','help','please']，发现单词 dog 果然出现了。同理，将其他元素值为 1 的都验证一遍，完全正确。

下面检查单词 dog 在第 5 篇文档中是否出现。

```
>>> setOfWordsVec(wordList,listOposts[4])
[0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
1, 0, 0, 0, 0, 0, 0, 1]
```

可以看到文档向量中索引为 0（对应词汇表中的 dog）的元素为 0，表示 dog 没有出现，可以查看第 5 篇文档['mr','licks','ate','my','steak','how','to','stop','him']，单词 dog 果然没有出现，完全正确。

18.4.4 训练算法

现在已经知道了一个单词在一篇文档中是否出现，也知道了该文档所属的类别。接下来重写贝叶斯公式，将之前的 x、y 替换为 **w**，粗体的 **w** 表示一个向量，即它由多个值组成。在这个例子中，数值个数与词汇表中的单词个数相同。

$$P(c_i \mid \mathbf{w}) = \frac{P(\mathbf{w} \mid c_i)P(c_i)}{p(\mathbf{w})}$$

通过前面的分析可以知道，需要计算每个类别的这个概率值，然后选取概率最大值对应的类别作为最终的类别。上式中每个类别概率的分母相等，因此只需要计算分子的两个概率 $P(c_i)$和 $P(\mathbf{w}|c_i)$。

首先通过类别 i（侮辱性留言或者非侮辱性留言）中的文档数除以总的文档数来计算概率 $P(c_i)$。接下来计算 $P(\mathbf{w}|c_i)$，这里假设所有单词都互相独立，将 **w** 展开为一个个独立的特征（单词），那么概率 $P(\mathbf{w}|c_i)$就可以写为 $P(w_0, w_1, w_2,\ldots, w_n|c_i)$，可以使用 $P(w_0|c_i)\ P(w_1|c_i)\ P(w_2|c_i)\ldots P(w_n|c_i)$来计算上述概率，这样计算过程就简便多了。

下面通过程序来计算两种类别下每个单词的概率 $P(w_i|c_0)$、$P(w_i|c_1)$和每个类别的概率值

$P(c_i)$。因为该问题是二分类问题，类别只有两类，因此只要计算一种类别的概率如 $P(c_1)$，另一种类别的概率 $P(c_0)=1-P(c_1)$。

$P(c_1)$表示带有侮辱性语言的文档的概率，可以用侮辱性文档数目除以文档总数得到。

$P(w_i|c_1)$表示侮辱性文档中单词 w_i 的条件概率，可以用侮辱性文档中单词 w_i 出现的次数除以所有侮辱性文档的单词总数得到。

为了简化程序的逻辑，在计算过程中用到了 Numpy 中的一些函数，故应确保导入 numpy 包，需要将 from numpy import *语句加在 Nbayes.py 文件的最前面。

```
from numpy import *
def trainNB(trainDocMatrix,trainCategory): #训练算法，由词向量计算概率
    numTrainDoc=len(trainDocMatrix)           #文档数
    numWord=len(trainDocMatrix[0])            #单词数
    #侮辱性文件的出现概率，即用 trainCategory 中所有的 1 的个数除以文档总数
    pAbusive=sum(trainCategory)/float(numTrainDoc)
    #构造单词出现的次数数组，初值为 0，大小为单词数
    p0Num=zeros(numWord)
    p1Num=zeros(numWord)
    #整个数据集单词出现的总数
    p0Denom=0.0
    p1Denom=0.0
    #对每个文档遍历
    for i in range(numTrainDoc):
        #是否为侮辱性文档
        if trainCategory[i]==1:
            #如果是侮辱性文档，对侮辱性文档的向量进行求和
            p1Num+=trainDocMatrix[i]
            #对向量中的所有元素求和，也就是计算所有侮辱性文档中出现的单词总数
            p1Denom+=sum(trainDocMatrix[i])
        else:
            p0Num+=trainDocMatrix[i]
            p0Denom+=sum(trainDocMatrix[i])
    #类别 1 下每个单词出现的概率
    p1Vec=p1Num/p1Denom
    #类别 0 下每个单词出现的概率
    p0Vec=p0Num/p0Denom
    return p0Vec,p1Vec,pAbusive
```

trainNB()函数有两个参数 trainDocMatrix 和 trainCategory，分别是所有训练文档的词向量构成的矩阵和文档对应的类别标签数组，输出为 $P(w|c_0)$（即 p0Vec）、$P(w|c_1)$（即 p1Vec）和 $P(c_1)$（即 pAbusive）。该函数首先计算 $P(c_1)$，即文档属于侮辱性文档（类别为 1）的概率，用侮辱性文档数除以文档总数即可得到。

计算 p0Vec 和 p1Vec（即 $P(w|c_0)$和 $P(w|c_1)$），需要计算每一个单词在每个类别下的概率，这里用了 Numpy 数组快速计算，p0Vec 和 p1Vec 都是向量，大小与词汇表相同，向量中的元素值表示在相应类别下对应的词汇表中的单词出现的概率。在程序中 p1Num 首先用

Numpy 的 zeros()初始化为长度与词汇表等长的全 0 数组；p1Denom 初始化为 0。在 for 循环中，遍历训练集的所有文档，如果文档是侮辱性文档，就将该文档和 p1Num 进行向量相加，并且将该文档的单词总数累加到 p1Denom；如果是正常文档，也做相应的处理。这样，循环结束，p1Num 向量中保存的是侮辱性文档中每个单词出现的次数；p1Denom 保存的是所有侮辱性文档中出现的单词总数，是一个整数；最后用每个单词出现的次数除以该类别中的单词总数（p1Num/p1Denom），则可以得到每个单词在该类别下的概率，即 $P(w_j|c_i)$（j=0,1,2,⋯, n)。

现在测试 trainNB()函数的效果。将 trainNB()函数的代码加入 Nbayes.py 文件中，运行该文件，并在 Python 提示符下输入：

```
>>> listOposts,listClasses=loadDataSet()
>>> wordList=vocabList(listOposts)
>>> wordList
```

至此生成一个包含所有词的词汇表 wordList，下面构建 trainNB()函数所需要的文档词向量矩阵。使用词向量生成函数 setOfWordsVec()循环生成每篇文档的词向量，填充到 trainDocMat 矩阵中，可以得到最终的文档词向量矩阵。

```
>>> trainDocMat=[]
>>> for inDoc in listOposts:
        trainDocMat.append(setOfWordsVec(wordList,inDoc))
>>> trainDocMat
[[1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 1, 0, 0, 0], [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1], [0, 0, 1, 0, 0, 0, 0, 0,
1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0],
[0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, 1, 0, 0, 0, 1, 0, 0], [0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1], [1, 0, 0, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]]
```

下面给出侮辱性文档的概率和两个类别的概率向量。

```
>>> p0V,p1V,pAb=trainNB(trainDocMat,listClasses)
>>> pAb
0.5
>>> p0V
array([0.04166667, 0.04166667, 0.125     , 0.04166667, 0.        ,
      0.04166667, 0.04166667, 0.04166667, 0.04166667, 0.04166667,
      0.04166667, 0.        , 0.04166667, 0.        , 0.        ,
      0.        , 0.04166667, 0.04166667, 0.        , 0.04166667,
      0.04166667, 0.        , 0.08333333, 0.        , 0.04166667,
      0.        , 0.        , 0.04166667, 0.04166667, 0.        ,
      0.04166667, 0.04166667])
>>> p1V
array([0.10526316, 0.05263158, 0.        , 0.        , 0.10526316,
```

```
        0.        , 0.        , 0.        , 0.        , 0.        ,
        0.        , 0.05263158, 0.        , 0.05263158, 0.05263158,
        0.05263158, 0.        , 0.        , 0.05263158, 0.        ,
        0.        , 0.05263158, 0.05263158, 0.05263158, 0.        ,
        0.15789474, 0.05263158, 0.        , 0.        , 0.05263158,
        0.        , 0.05263158])
```

可以看到，侮辱性文档的概率 pAb 为 0.5，从训练数据可知，一共有 6 篇文档，其中 3 篇是侮辱性文档，因此该值正确。下面看看单词在给定类别下的概率是否正确，词汇表的第 1 个单词是 dog，其在类别 0 中出现 1 次，而类别 0 的文档单词总数为 24 个，对应的条件概率为 1/24=0.04166667；在类别 1 中出现两次，类别 1 的文档单词总数为 19 个，对应的条件概率为 2/19=0.10526316，该计算是正确的。下面看看两类概率中的最大值，发现是 p1V 中的 0.15789474，该概率的索引为 25，对应词汇表中的 stupid，意味着该单词 stupid 是最能表征类别（侮辱性文档类）的单词。通过查看训练数据，可以发现单词 stupid 在 3 篇侮辱性文档中全部出现了。

18.4.5　测试算法并改进

从上面的测试结果可以看到，在 p0V 和 p1V 中都存在一些单词概率为 0 的现象，那么计算 $P(w_0|c_i)P(w_1|c_i)P(w_2|c_i)\cdots P(w_n|c_i)$的值肯定为 0，这将导致用朴素贝叶斯分类器对文档分类时下面式子中的分子为 0，无法比较概率大小，最终无法分类。

$$P(c_i\,|\,\mathbf{w}) = \frac{P(\mathbf{w}\,|\,c_i)P(c_i)}{p(\mathbf{w})}$$

为了避免这种现象，可以将所有单词的出现次数由初始化为 0 改为 1，并将 p1Vec= p1Num/p1Denom 和 p0Vec = p0Num/p0Denom 中的分母初始化为 2.0，确保分母不为 0。注意，这里初始化为 1 或 2 的目的主要是为了保证分子和分母不为 0，大家可以根据业务需求进行更改。

将 trainNB()函数中初始化的这几条语句：

```
p0Num=zeros(numWord)
p1Num=zeros(numWord)
p0Denom=0.0
p1Denom=0.0
```

修改为：

```
p0Num=ones(numWord)
p1Num=ones(numWord)
p0Denom=2.0
p1Denom=2.0
```

另一个需要注意的问题是下溢出，这是由于很多很小的数相乘造成的。当计算乘积 $P(w_0|c_i)P(w_1|c_i)P(w_2|c_i)\cdots P(w_n|c_i)$时，由于大部分因子都非常小，乘积将会变得更小，所以可能会产生下溢出或者得到不正确的答案。一种有效的解决办法是对乘积取自然对数。因为

ln(ab)=ln(a)+ln(b)，于是通过求对数可以避免下溢出或者浮点数舍入导致的错误。在数学上，f(x)与 ln(f(x))的取值结果虽然不同，但在相同区域内变化的趋势一致，并且在相同值上取到极值。因此，采用自然对数进行处理不会影响最终结果，所以将 trainNB()函数中 return 前的两行代码修改为：

```
p1Vec=log(p1Num/p1Denom)
p0Vec=log(p0Num/p0Denom)
```

到现在为止，分类器已经修改好了，下面开始检验分类效果。

18.4.6　使用算法进行文本分类

依据朴素贝叶斯公式 $P(c_i\,|\,\mathbf{w})=\dfrac{P(\mathbf{w}\,|\,c_i)P(c_i)}{p(\mathbf{w})}$ 及前面的分析，只要计算公式中的分子即可，重点是比较分子的大小，分子大的对应的类别就是文档的类别。将 trainNB()函数求得的 3 个概率 p0Vec、p1Vec、pAbusive 及需要分类文档的词向量代入公式，分别比较两种类别下的概率大小，即可确认文档的类别。classifyNB()函数实现了上述过程。

```
def classifyNB(textVec,p0Vec,p1Vec,pClass1):
    """
    分类函数
    :param textVec: 要分类的文档向量
    :param p0Vec: 正常文档类（类别 0）下的单词概率列表
    :param p1Vec: 侮辱性文档类（类别 1）下的单词概率列表
    :param pClass1: 侮辱性文档（类别 1）概率
    :return: 类别 1 或 0
    """
    p1=sum(textVec*p1Vec)+log(pClass1)
    p0=sum(textVec*p0Vec)+log(1.0-pClass1)
    print('p1=',p1)
    print('p0=',p0)
    if p1>p0:
        return 1
    else:
        return 0
```

为了测试更加方便，将前面在 Python 提示符下的所有操作进行封装，构造测试函数 testNB()。

```
def testNB():                    #朴素贝叶斯算法测试
    #(1)加载数据集
    listOposts,listClasses=loadDataSet()
    #(2)创建词汇表
    wordList=vocabList(listOposts)
```

```
    #(3)构造训练数据的文档词向量矩阵
    trainDocMat=[]
    for inDoc in listOposts:
        trainDocMat.append(setOfWordsVec(wordList, inDoc))
    #(4)训练数据
    p0V,p1V,pAb=trainNB(array(trainDocMat),array(listClasses))
    #(5)测试数据
    testText=['love','my','ate']
    thisDoc=array(setOfWordsVec(wordList,testText))
    print(testText,'classified as:',classifyNB(thisDoc,p0V,p1V,pAb))
    testText=['stupid','dog']
    thisDoc=array(setOfWordsVec(wordList,testText))
    print(testText,'classified as:',classifyNB(thisDoc,p0V,p1V,pAb))
```

将所有函数保存到 Nbayes.py 文件中，运行该文件，并在 Python 提示符下输入：

```
>>> testNB()
```

可以看到运行结果如下：

```
p1=-9.826714493730215
p0=-7.694848072384611
['love', 'my', 'ate'] classified as: 0
p1=-4.2972854062187915
p0=-6.516193076042964
['stupid', 'dog'] classified as: 1
```

18.5　使用朴素贝叶斯分类算法过滤垃圾邮件

视频讲解

18.4 节运用朴素贝叶斯算法完整实现了社区留言板言论的分类，下面将朴素贝叶斯算法应用于垃圾邮件过滤，同样用朴素贝叶斯算法分类的通用框架来解决该问题。使用朴素贝叶斯算法对电子邮件进行分类的实现流程如下。

（1）收集训练数据：提供已知的文本文件。

（2）准备数据：将文本文件解析为词向量。

（3）分析数据：检查词条确保解析的正确性。

（4）训练算法：从词向量计算概率，使用前面建立的 trainNB()函数。

（5）测试算法：使用朴素贝叶斯算法进行交叉验证。

（6）使用算法：构建完整的程序对一组邮件进行分类，将错分的邮件输出到屏幕上。

18.5.1　收集训练数据

假设训练数据已知，这里用 50 封邮件作为训练数据，其中垃圾邮件和非垃圾邮件各 25 封。

18.5.2 将文本文件解析为词向量

在 18.4 节中，为了着重理解朴素贝叶斯算法分类的核心过程，文本的词向量是预先给定的，下面介绍如何从文本构建自己的词向量。

对于一个英文文本字符串，可以使用 Python 的 string.split()方法切分。下面看一下实际的运行效果。在 Python 提示符下输入：

```
>>> text='In fact, persistence hunting remained in use until 2014, such
as with the San people of the Kalahari Desert.'
>>> text.split()
['In', 'fact,', 'persistence', 'hunting', 'remained', 'in', 'use', 'until',
'2014,', 'such', 'as', 'with', 'the', 'San', 'people', 'of', 'the',
'Kalahari', 'Desert.']
```

可以看到在默认情况下split()按照单词之间的空格进行切分，整体切分效果不错，但美中不足的是，标点符号也被当作单词的一部分进行了切分。可以使用正则表达式来切分文本，其中分隔符是除单词、数字以外的任意字符串。

```
>>> import re
>>> regEx=re.compile('\W*')
>>> wordList=regEx.split(text)
>>> wordList
['In', 'fact', 'persistence', 'hunting', 'remained', 'in', 'use', 'until',
'2014', 'such', 'as', 'with', 'the', 'San', 'people', 'of', 'the', 'Kalahari',
'Desert', '']
```

可以看到标点符号没有了，但是多了空字符串，可以计算字符串的长度，只返回长度大于 0 的字符串，去掉空字符串。这里用列表推导式实现：

```
>>> [word for word in wordList if len(word)>0]
['In', 'fact', 'persistence', 'hunting', 'remained', 'in', 'use', 'until',
'2014', 'such', 'as', 'with', 'the', 'San', 'people', 'of', 'the', 'Kalahari',
'Desert']
```

可以看到空字符串去掉了，但是英文句子的第 1 个单词和专用名词的首字母是大写的，为了使所有单词的形式统一，将其全部转换成小写，用 lower()函数即可实现。

```
>>> [word.lower() for word in wordList if len(word)>0]
['in', 'fact', 'persistence', 'hunting', 'remained', 'in', 'use', 'until',
'2014', 'such', 'as', 'with', 'the', 'san', 'people', 'of', 'the', 'kalahari',
'desert']
```

另外，在实际处理中通常也要过滤掉长度小于 3 的字符串，使得词汇表尽量小一些。

将上面的文本处理过程整理为一个独立的文本解析函数。

```
import re
```

```
def textParse(text):
    regEx=re.compile('\W*')
    wordList=re.split(regEx,text)
    return [word.lower() for word in wordList if len(word)>2]
```

当然，在实际应用中文本解析是一个相当复杂的过程，这里是一种简单的处理。尤其是中文文本的解析，由于不像英文句子中有空格分隔，要识别出中文文本中的词汇，即中文分词本身就是一个值得研究的应用领域。但好在现在有一些成熟的支持 Python 的专门的分词模块，这里推荐使用 jieba 分词，它专门使用 Python 的分词系统，占用的资源少，常识类文档的分词精度较高，对于非专业文档绰绰有余，如果需要可以直接下载使用。

现在将电子邮件文本传入 textParse()函数，就可以得到该电子邮件的单词列表。

调用 18.4 节实现的 vocabList()函数可以生成所有训练数据的词汇表。构建文档的词向量可以用 18.4 节实现的 setOfWordsVec()函数。训练算法则直接调用 trainNB()函数，这里不再重复叙述。直接用分类算法进行邮件分类测试。

18.5.3　使用朴素贝叶斯算法进行邮件分类

这里使用朴素贝叶斯算法对邮件进行分类，并进行交叉验证。交叉验证也称为循环估计，是一种统计学上将数据样本切割成较小子集的实用方法。在给定的建模样本中，拿出大部分样本进行建模，留小部分样本用刚建立的模型进行预报，并求这小部分样本的预报误差。

本例中的样本数据共有 50 封邮件，随机选择 10 封邮件作为测试集，剩下的 40 封邮件作为训练集。

```
def spamTest():
    """
    对贝叶斯垃圾邮件分类器进行自动化处理
    return:对测试集中的每封邮件进行分类，若邮件分类错误，则错误数加 1，最后返回错分率
    """
    emailWordList=[]            #邮件的词向量列表，大小与邮件数相同
    classList=[]                #邮件的类别标签列表
    #这里提供的训练邮件共 50 封，垃圾邮件和正常邮件分别 25 封
    for i in range(1,26):
        #切分，解析数据，并归类为 1 类别
        wordList=textParse(open('email/spam/%d.txt' % i ,encoding="utf-8")
        .read())
        emailWordList.append(wordList)
        classList.append(1)
        #切分，解析数据，并归类为 0 类别
        wordList=textParse(open('email/noSpam/%d.txt' % i ,encoding="utf-8")
        .read())
        emailWordList.append(wordList)
        classList.append(0)
```

```
    #创建词汇表
    wordTable=vocabList(emailWordList)
    trainingSet=list(range(50))
    #构造测试集
    testSet=[]
    #随机选择 10 封邮件用来测试
    for i in range(10):
        randIndex=int(random.uniform(0,len(trainingSet)))
        testSet.append(trainingSet[randIndex])
        del(trainingSet[randIndex])
    #构造训练集文档词向量矩阵和对应的类别向量
    trainDocMat=[]
    trainDocClass=[]
    for docIndex in trainingSet:
         trainDocMat.append(setOfWordsVec(wordTable,
          emailWordList[docIndex]))
         trainDocClass.append(classList[docIndex])
    #用训练集的邮件进行训练
    p0V,p1V,pSpam=trainNB(array(trainDocMat),array(trainDocClass))
    errorCount=0          #记录错误邮件的数目
    #用测试集中的邮件进行分类测试
    for docIndex in testSet:
        #对每一封邮件生成词向量
        wordVector=setOfWordsVec(wordTable,emailWordList[docIndex])
        #测试的分类类别与原标注类别比较，若不相等，说明分类错误
        if classifyNB(array(wordVector),p0V,p1V,pSpam)!=
        classList[docIndex]:
            print("classification error:",docIndex)
            errorCount+=1
    errRate=float(errorCount/len(testSet))      #错分率
    print('the error rate is :', errRate)
```

spamTest()函数对朴素贝叶斯垃圾邮件分类器进行自动化处理。首先导入已经整理好的垃圾邮件和正常邮件的文本文件，分别在文件夹 spam 与 noSpam 下，并分别对它们进行解析处理，生成邮件的词向量列表 emailWordList 和类别列表 classList。调用 18.4 节案例的 vocabList()方法，可以得到没有重复单词的词汇表 wordTable。

接下来进行交叉验证，需要分别构建训练集和测试集，本例中共有 50 封邮件，随机选择 10 封作为测试集，剩下的 40 封邮件作为训练集。那么如何选择呢？初始化时 trainingSet= list(range(50))、testSet=[]，可以知道 trainingSet 是一个 0～49 的整数列表，testSet 是一个空列表。随机函数 random.uniform()可以生成一个指定范围内的随机浮点数，int(random.uniform(0,len(trainingSet)))将产生一个 0～49 的整数，产生的数字加入测试集列表 testSet 中，同时将该数字从训练集 trainingSet 中删除，循环 10 次，就随机产生了 10 个 0～49 的整数，整数索引对应的邮件作为测试集，剩下的 40 封邮件作为训练集。

接着遍历训练集中的所有文档，对每封邮件基于词汇表并使用 setOfWordsVec()函数构

造训练集的词向量矩阵 trainDocMat，同时生成类别向量 trainDocClass，并通过训练函数 trainNB()计算出分类所需要的 3 个概率。

最后遍历测试集，对每封邮件用 classifyNB()进行分类，与已知的邮件类别进行对比，如果分类错误则错误数加 1，并输出错分的邮件索引，以便于进一步查验，最后给出总的错分率。

下面对上述过程进行测试，将上述所有程序代码保存至 Nbayes.py 文件中并运行，在 Python 提示符下输入：

```
>>> spamTest()
the error rate is : 0.0
>>> spamTest()
classification error: 32
the error rate is : 0.1
```

spamTest()函数会输出 10 封随机选择的邮件的分类错误率，上面的测试是运行两次的结果，因为测试集的 10 封邮件是随机选择的，每次选择的测试集不一定相同，所以每次的运行结果可能会有些差别。如果有分类错误，会输出文档索引，以便于进一步分析错分的邮件。为了更精确地估计错分率，需要重复运行多次，然后求平均值，运行 100 次，获得的平均错分率为 3.5%。

18.5.4　改进算法

在前面的算法中，把邮件文本用词集模型表示，将文本的每个词作为一个特征，将词是否出现作为特征的值。实际上，大家经常见到一个词在一个文本中多次出现的情况，一个词多次出现是否可以比是否出现更加能够表达某种意义？这种方法被称为词袋（Bag of Words）模型。下面来试一试。

只要将词集模型中的单词是否出现修改为出现次数就可以了，这样只需要对 setOfWordsVec()函数做简单修改，当扫描到一个单词时就将词向量中该单词的对应值加 1。

```
def bagOfWordsVec(vocabList,inputText):
    textVec=[0]*len(vocabList)       #创建一个所包含元素都为 0 的向量
    #遍历文档中的所有单词，若出现了词汇表中的单词，则将文档向量中的对应值加 1
    for word in inputText:
      if word in vocabList:
          textVec[vocabList.index(word)]+=1
    return textVec
```

修改 spamTest()函数，将两处生成词向量的语句改为调用上面的词袋模型函数 bagOfWordsVec()，重新执行 spamTest()函数 100 次，得到的错分率为 0，看来这样修改后确实可以提高分类的准确度。当然，在实际分类中是做不到百分之百正确的，这可能和用户选用的数据集有关。

18.6 使用 Scikit-Learn 库进行文本分类

Scikit-Learn 是一个用于机器学习的 Python 库，建立在 Numpy、Scipy 和 Matplotlib 基础之上。它提供了机器学习常用的算法模块，例如监督学习、无监督学习、模型选择和评估、数据集转换等，在监督学习模块中包含机器学习常用的分类算法，例如朴素贝叶斯、KNN、决策树、支持向量机等。

Scikit-Learn 的官方网站“http://scikit-learn.org”是学习和应用机器学习算法的最重要的工具之一，以朴素贝叶斯算法为例，网站提供了一整套算法学习的教程和资源，网址为“http://scikit-learn.org/stable/modules/naive_bayes.html”。

如果要安装 Scikit-Learn，需要先安装 Numpy、Scipy 和 Matplotlib，直接用 pip install 命令安装：

```
pip install numpy
pip install scipy
pip install matplotlib
pip install scikit-learn
```

本节选择 Scikit-Learn 的朴素贝叶斯算法进行文本分类，对 18.4 节和 18.5 节的例子重新进行实现。

用 Scikit-Learn 的朴素贝叶斯算法进行文本分类包含以下几个步骤：

（1）收集样本数据。

（2）提取文本特征：生成文本的向量空间模型。

（3）训练分类器。

（4）在测试集上进行测试。

（5）分类结果评估。

18.6.1 文本分类常用的类和函数

视频讲解

❶ load_files()函数

该函数位于 sklearn.datasets 模块下，功能是加载文本文件，将二层文件夹名字作为分类类别。Scikit-Learn 本身也自带了一些数据集，可以供用户学习测试使用，一般通过 sklearn.datasets 模块下的其他函数加载使用。

```
sklearn.datasets.load_files(container_path, description=None,
categories=None, load_content=True, shuffle=True, encoding=None,
decode_error='strict', random_state=0)
```

主要参数如下。

- container_path：文件夹的路径。
- load_content：是否把文件中的内容加载到内存，该项为可选项，默认值为 True。

- encoding：编码方式。当前文本文件的编码方式一般为“utf-8”，如果不指明编码方式（encoding=None），那么文件内容将会按照 bytes 处理，而不是按照 unicode 处理，其默认值为 None。

该函数的返回值为 Bunch 对象，主要属性如下。

- data：原始数据。
- filenames：每个文件的名字。
- target：类别标签（从 0 开始的整数索引）。
- target_names：类别标签的具体含义（由子文件夹的名字决定）。

注意：需要将数据组织成如下文件结构，有几种类别就有几个子文件夹，当然文件夹及文件的名字可以自定义。

```
container folder/
    category 1 folder/
        file 1.txt file 2.txt ... file 42.txt
    category 2 folder/
        file_43.txt file_44.txt ...
```

例如在垃圾邮件过滤例子中，email 文件夹下的二级文件夹 spam 和 noSpam 将作为类别标签。

❷ **train_test_split()函数**

该函数位于 sklearn.model_selection 模块下，能够从样本数据随机按比例选取训练子集和测试子集，并返回划分好的训练集测试集样本和训练集测试集标签。

```
sklearn.model selection.train test split(train data,train target,
test_size, random_state)
```

参数如下。

- train_data：被划分的样本特征集。
- train_target：被划分的样本标签。
- test_size：如果是 0～1 的浮点数，表示样本占比；如果是整数，则是样本的数量，该项为可选项，默认值为 None。
- random_state：随机数的种子，不同的种子会造成不同的随机采样结果，相同的种子采样结果相同，该项为可选项，默认值为 None。

❸ **CountVectorizer 类**

该类是文本特征提取模块 sklearn.feature_extraction.text 下的一个常用的类，能够将文档词块化，并进行数据预处理，例如去音调、转小写、去停用词，最后生成文档的词频矩阵，即前面所说的词袋模型。

```
class sklearn.feature extraction.text.CountVectorizer(input=u'content',
encoding=u'utf-8', decode error=u'strict',strip accents=None, lowercase
=True, preprocessor=None, tokenizer=None, stop words=None,token pattern
=u'(?u)\b\w\w+\b', ngram_range=(1, 1), analyzer=u'word', max_df=1.0,min
 df=1,max features=None, vocabulary=None, binary=False, dtype=<type 'numpy.
int64'>)
```

其参数很多，这里介绍最常用的几个，其他用默认值即可。

- stop_words：设置停用词，可以为 english、list 或 None（默认值），设为 english 将使用内置的英语停用词，设为一个 list 可自定义停用词，设为 None 则不使用停用词，设为 None 且 max_df∈[0.7, 1.0)将自动根据当前的语料库建立停用词表。
- lowercase：是否将所有字符转变成小写，默认值为 True。
- token_pattern：表示 token 的正则表达式，只有当 analyzer == 'word'时才使用，默认的正则表达式选择两个及以上的字母或数字作为 token，标点符号默认当作 token 分隔符，而不会被当作 token。
- analyzer：一般使用默认值，可设置为 string 类型{'word', 'char', 'char_wb'}或 callable，特征基于 wordn-grams 或 character n-grams。
- decode_error：默认为 strict，若遇到不能解码的字符将报 UnicodeDecodeError 错误，设为 ignore 将会忽略解码错误。

CountVectorizer 类的核心函数是 fit_transform()，通过该函数能够学习词汇表，返回文档的词频矩阵。

```
fit_transform(raw_documents, y=None)
```

其参数 raw_documents 为迭代器，可以是 str、unicode 或文件对象。

其返回值是文档的词频矩阵，矩阵大小为文档数×特征数。

❹ MultinomialNB 类

在 Scikit-Learn 中共有 3 个朴素贝叶斯的分类算法类，分别是 GaussianNB、MultinomialNB 和 BernoulliNB。其中，GaussianNB 是先验概率为高斯分布的朴素贝叶斯，MultinomialNB 是先验概率为多项式分布的朴素贝叶斯， BernoulliNB 是先验概率为伯努利分布的朴素贝叶斯。

这 3 个类适用的分类场景不同，一般来说，如果样本特征的分布大部分是连续值，使用 GaussianNB 会比较好；如果样本特征的分布大部分是多元离散值，使用 MultinomialNB 比较合适；如果样本特征是二元离散值或者很稀疏的多元离散值，应该使用 BernoulliNB。因为考虑文本分类中的特征是离散的单词，所以用 MultinomialNB。

```
class sklearn.naive bayes.MultinomialNB(alpha=1.0,fit prior=True,class prior=None)
```

参数如下。

- alpha：拉普拉斯平滑参数，是一个大于 0 的常数，该项为可选项，默认值为 1。
- fit_prior：是否要考虑先验概率，如果是 False，则所有的样本类别输出都有相同的类别先验概率，该项为可选项，默认值为 True。
- class_prior：类别的先验概率，该项为可选项，默认值为 None。

MultinomialNB 类下有两个核心函数。

（1）fit()函数：拟合朴素贝叶斯分类器。

```
fit(X, y, sample_weight=None)
```

参数如下。

- X：训练数据的词频矩阵。
- y：训练数据的类别标签向量。

其返回值为 MultinomialNB 对象。

（2）predict()函数：对测试集进行分类。

```
predict(X)
```

参数 X 为要分类的文档的词频矩阵，矩阵大小为测试集中的文档数×特征数。

其返回值为 X 的预测目标值向量，向量大小与测试集中的文档数一致。

❺ classification_report()函数

该函数位于 sklearn.metrics 模块下，用于显示主要分类指标的文本报告，在报告中显示每个类的准确率、召回率、F1 值等信息。

```
sklearn.metrics.classification report(y true, y pred, labels=None,
target_names=None, sample_weight=None, digits=2)
```

主要参数如下。

- y_true：一维数组，或标签指示器数组/稀疏矩阵，目标值。
- y_pred：一维数组，或标签指示器数组/稀疏矩阵，分类器返回的估计值。
- labels：一维数组，报表中包含的标签索引列表，可选。
- target_names：字符串列表，与标签匹配的显示名称，可选。
- sample_weight：一维数组，样本权重，可选。
- digits：int 型，输出浮点值的位数，可选。

其返回值为每类文本的准确率、召回率、F1-score 等信息，string 类型。

对于数据测试结果有下面 4 种情况。

TP：预测为正，实际为正；FP：预测为正，实际为负；FN：预测为负，实际为正；TN：预测为负，实际为负。

关于准确率、召回率、F1-score，详细定义如下：

准确率=TP/ (TP+FP)

召回率=TP/ (TP + FN)

F1−score= 2×TP/(2×TP + FP + FN)

熟悉了这些类和函数的使用，就可以实现具体的案例了。

18.6.2　案例实现

创建一个名为 sklearn_NB.py 的新文件，用 Scikit-Learn 库的朴素贝叶斯算法将 18.4 节和 18.5 节的例子重新进行实现。

```
from sklearn import datasets
from sklearn.feature extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB      #导入多项式贝叶斯算法包
from sklearn.cross_validation import train_test_split
```

```
from sklearn.metrics import classification_report
#留言板案例的 Scikit-Learn 实现
def testNB_skl():
    posting=['my dog has flea problems help please','maybe not take him to
             dog park stupid',
             'my dalmation is so cute I love him','stop posting stupid
             worthless garbage',
             'mr licks ate my steak how to stop him','quit buying worthless
             dog food stupid']
    classVec=[0,1,0,1,0,1]
    #交叉验证选择训练集和测试集
    train_data,test_data,train_y,test_y=train_test_split(posting,
    classVec,test_size=0.2,train_size=0.8)
    #生成文本的词频矩阵
    vectorizer=CountVectorizer()    #CountVectorizer 用于词袋模型统计词频
    wordX=vectorizer.fit_transform(train_data)
    #训练分类器
    clf=MultinomialNB().fit(wordX,train_y)
    #预测测试集的分类结果
    test_wordX=vectorizer.transform(test_data).toarray()
    predicted=clf.predict(test_wordX)          #预测
    for doc,category in zip(test_data,predicted):
        print(doc,":",category)
    #在测试集上的性能评估
    classTarget_names=['正常言论','侮辱性言论']
    print(classification_report(test_y,predicted,target_names=
    classTarget_names))
#垃圾邮件过滤的 Scikit-Learn 实现
def spamTest_skl():
    #加载 email 文件夹下的数据
    base_data=datasets.load_files("email/")
    #交叉验证选择训练集和测试集
    train_data,test_data,train_y,test_y=
    train_test_split(base_data.data,base_data.target,test_size=
    0.2,train_size=0.8)
    #生成文本的词频矩阵
    vectorizer=CountVectorizer(stop_words="english",
    decode_error='ignore')
    wordX=vectorizer.fit_transform(train_data)
    #训练分类器
    clf=MultinomialNB().fit(wordX,train_y)
    #预测测试集的分类结果
    test_wordX=vectorizer.transform(test_data).toarray()
    #newDoc_tfidf=transformer.transform(newDoc_wordX)
                                               #得到新文档每个词的 TF-IDF 值
    predicted=clf.predict(test_wordX)          #预测
```

```
    print(predicted)
    #在测试集上的性能评估
    print(classification_report(test_y,predicted,target_names=
    base_data.target_names))
```

运行 sklearn_NB.py，在 Python 提示符下输入 testNB_skl()，可以得到留言是否为侮辱性言论的预测分类结果及预测评估报告：

```
maybe not take him to dog park stupid : 0
quit buying worthless dog food stupid : 1
             precision    recall   f1-score    support
  正常言论       0.00        0.00       0.00         0
  侮辱性言论     1.00        0.50       0.67         2
avg/total       1.00        0.50       0.67         2
```

在 Python 提示符下输入 spamTest_skl()，可以得到垃圾邮件过滤的预测分类结果及预测评估报告：

```
[0 1 0 0 1 1 0 0 0 1]
           precision    recall   f1-score   support
     noSpam    1.00        1.00      1.00        6
       spam    1.00        1.00      1.00        4
avg/total      1.00        1.00      1.00       10
```

注意：因为训练集和测试集是随机划分的，所以每次的运行结果不一定相同。

第19章 深度学习案例——基于卷积神经网络的手写体识别

19.1 手写体识别案例需求

视频讲解

人类对图 19-1 所示的一串手写图像可以毫不费力地认出是 504192，这是因为人体的视觉系统相当神奇，但是让计算机进行识别就比较复杂了。假如给定一个数字 5 的图像，计算机如何描述出这是一个数字 5 呢？我们可以把计算机当作一个小孩子，让它见很多的 5 的图片，慢慢就形成了自己的判断标准，而这种让计算机学习的方法就是神经网络，深度学习（Deep Learning）就是具有多隐含层的神经网络结构。



图 19-1　手写体数字

本章案例将采用深度学习框架，使用卷积神经网络（CNN）对 MNIST 数据集进行训练，最终给计算机一个任意书写的手写体数字，使它能够识别出该数字是什么。

19.2 深度学习的概念及关键技术

深度学习（Deep Learning）是机器学习（Machine Learning）研究中的一个新领域，是具有多隐含层的神经网络结构。

19.2.1 神经网络模型

❶ 生物神经元

大脑大约由 140 亿个神经元组成，神经元互相连接成神经网络，每个神经元平均连接几千条其他神经元。神经元是大脑处理信息的基本单元。一个神经元的结构如图 19-2 所示。

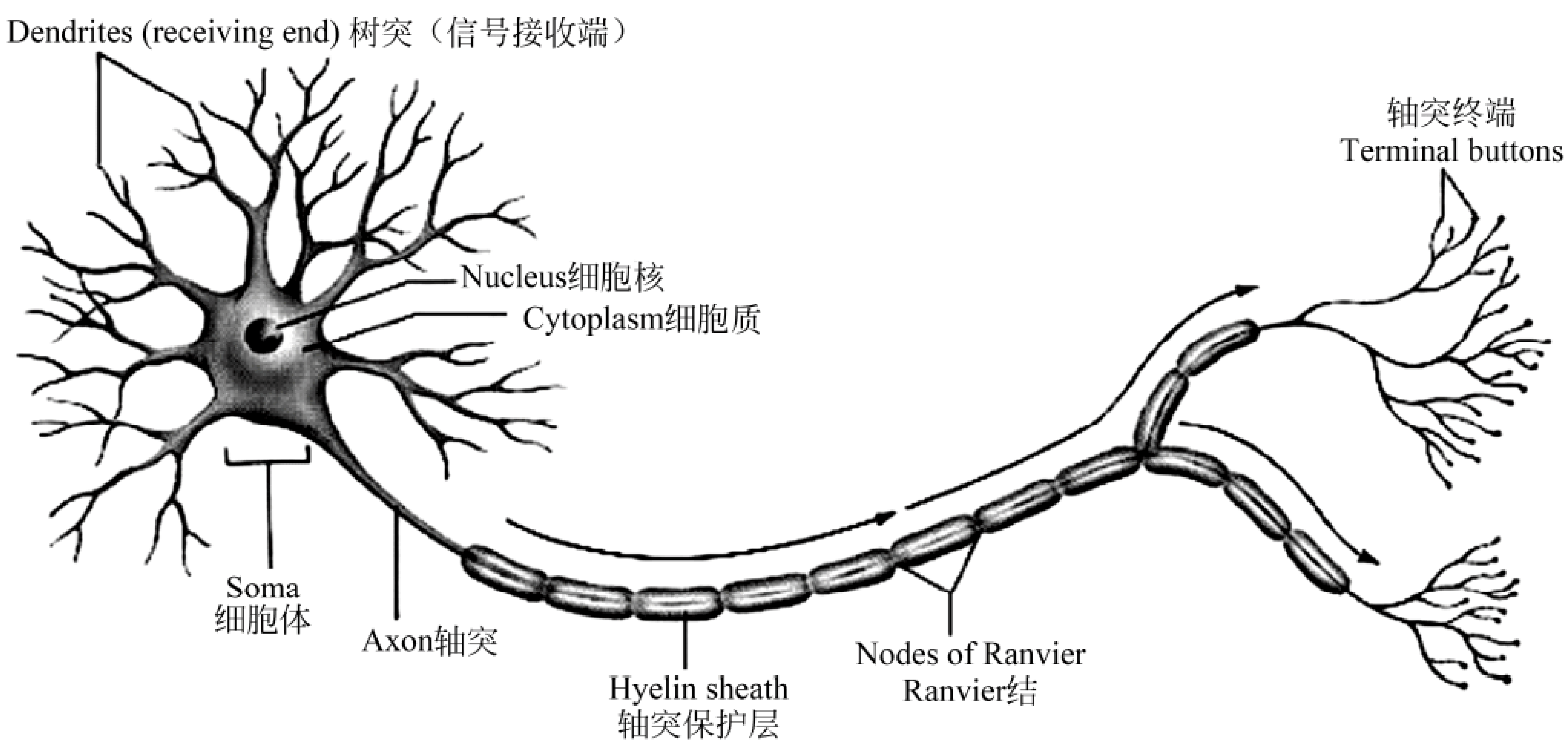


图 19-2　生物神经元结构

可以看到，一个可视化的生物神经元是由细胞体、树突和轴突三部分组成。以细胞体为主体，由许多向周围延伸的不规则树枝状纤维构成，其形状像一棵枯树的枝干。其中，轴突负责细胞体到其他神经元的输出连接，树突负责接收其他神经元到细胞体的输入。来自神经元（突触）的电化学信号聚集在细胞核中，如果聚合超过了突触阈值，那么电化学尖峰（突触）就会沿着轴突向下传播到其他神经元的树突上。

由于神经元结构的可塑性，突触的传递作用可增强或减弱，因此神经元具有学习与遗忘的功能。

❷ 人工神经网络

人工神经网络是反映人脑结构及功能的一种抽象数据模型，它使用大量的人工神经元进行计算，该网络将大量的“神经元”相互连接，每个“神经元”是一种特定的输出函数，又称为激活函数。每两个“神经元”之间的连接都通过加权值，称为权重，这相当于人工神经网络的记忆。网络的输出则根据网络的连接规则来确定，输出因权重值和激励函数的不同而不同。

一个简单的人工神经网络如图 19-3 所示，其中，$x_1(t)$等数据为这个神经元的输入，代表其他神经元或外界对该神经元的输入；ω_{i1} 等数据为这个神经元的权重，$u_i = \sum_j \omega_{ij} \cdot x_j(t)$ 是对输入的求和；$y_i(t) = f(u_i(t))$ 称为激励函数，是对求和部分的再加工，也是最终的输出。

因此，神经网络就是将许多单一的神经元连接在一起的一个典型的网络，如图 19-4 所示，用更多的神经元去进行学习，神经网络最左边的一层叫输入层，它有 3 个输入单元；最右边的一层叫输出层，它只有一个结点；中间两层称为隐藏层，因为用户不能在训练过程中观测到它们的值。其实，神经网络可以包含更多的隐藏层。

19.2.2　深度学习之卷积神经网络

视频讲解

深度学习的概念源于人工神经网络的研究，含有多隐层的神经网络就是一种深度学习结构。深度学习通过组合低层特征形成更加抽象的高层表

示属性类别或特征，以发现数据的分布式特征表示。

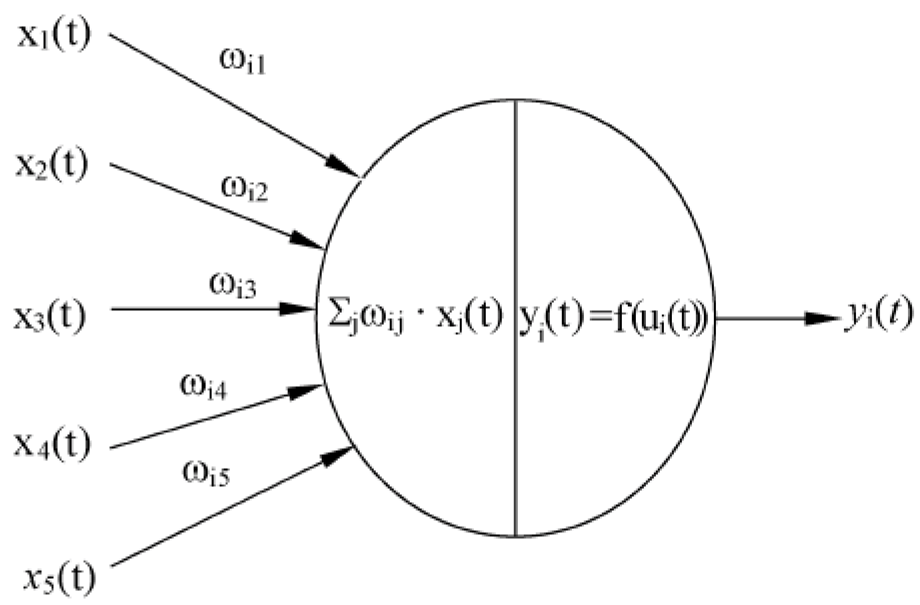


图 19-3　人工神经网络

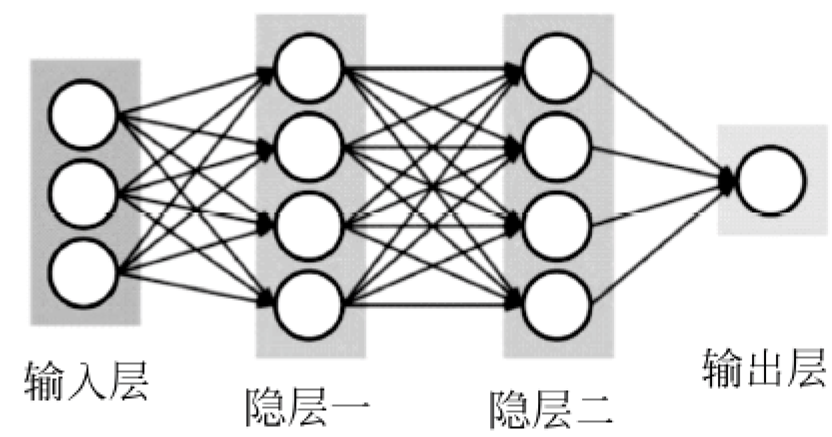


图 19-4　神经网络典型结构

深度学习中的卷积神经网络（CNN）近年来有了非常出色的表现，它与普通的神经网络的区别在于包含了一个由卷积层和池化层构成的特征抽取器。在卷积神经网络的卷积层中，一个神经元只与部分邻层神经元相连接，通常包含若干个特征图（Feature Map），每个特征平面由一些矩形排列的神经元组成。同一特征平面的神经元共享权值，这里共享的权值就是卷积核，卷积核一般以随机小数矩阵的形式初始化，在网络的训练过程中卷积核将学习得到合理的权值。共享权值（卷积核）带来的直接好处是减少了网络各层之间的连接，同时又降低了过拟合的风险。池化也叫子采样（Pooling），可以看作一种特殊的卷积过程。卷积和池化大大简化了模型复杂度，减少了模型的参数。

下面具体介绍几个相关概念。

❶ 卷积

这里用一个简单的例子来讲述如何计算卷积，假设有一个 5×5 的图像，使用一个 3×3 的卷积核（filter）进行卷积，想得到一个 3×3 的 Feature Map，首先对图像的每个像素进行编号，用 $x_{i,j}$ 表示图像的第 i 行第 j 列元素，对 filter 的每个权重进行编号，用 $w_{m,n}$ 表示第 m 行第 n 列的权重，对 Feature Map 的每个元素进行编号，用 $a_{i,j}$ 表示第 i 行第 j 列元素。

那么 Feature Map 中 $a_{0,0}$ 的卷积计算方法如下，如图 19-5 所示。

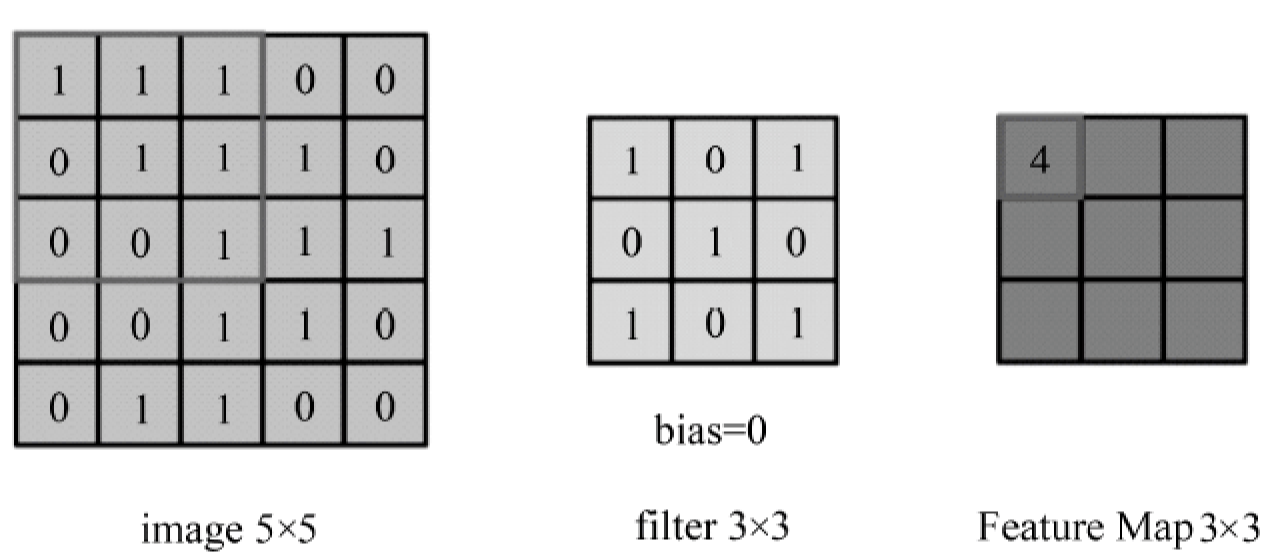


图 19-5　卷积原理图 1

$$
\begin{aligned}
a_{0,0} &= \omega_{0,0}x_{0,0}+\omega_{0,1}x_{0,1}+\omega_{0,2}x_{0,2}+\omega_{1,0}x_{1,0}+\omega_{1,1}x_{1,1}+\omega_{1,2}x_{1,2}+\omega_{2,0}x_{2,0}+\omega_{2,1}x_{2,1}+\omega_{2,2}x_{2,2} \\
&= 1\times1+0\times1+1\times1+0\times0+1\times1+0\times1+1\times0+0\times0+1\times1 \\
&= 4
\end{aligned}
$$

Feature Map 中 $a_{0,1}$ 的卷积计算方法如下，如图 19-6 所示。

$$a_{0,1}=\omega_{0,1}x_{0,1}+\omega_{0,2}x_{0,2}+\omega_{0,3}x_{0,3}+\omega_{1,1}x_{1,1}+\omega_{1,2}x_{1,2}+\omega_{1,3}x_{1,3}+\omega_{2,1}x_{2,1}+\omega_{2,2}x_{2,2}+\omega_{2,3}x_{2,3}$$
$$=1\times1+0\times1+1\times0+0\times1+1\times1+0\times1+1\times0+0\times1+1\times1$$
$$=3$$

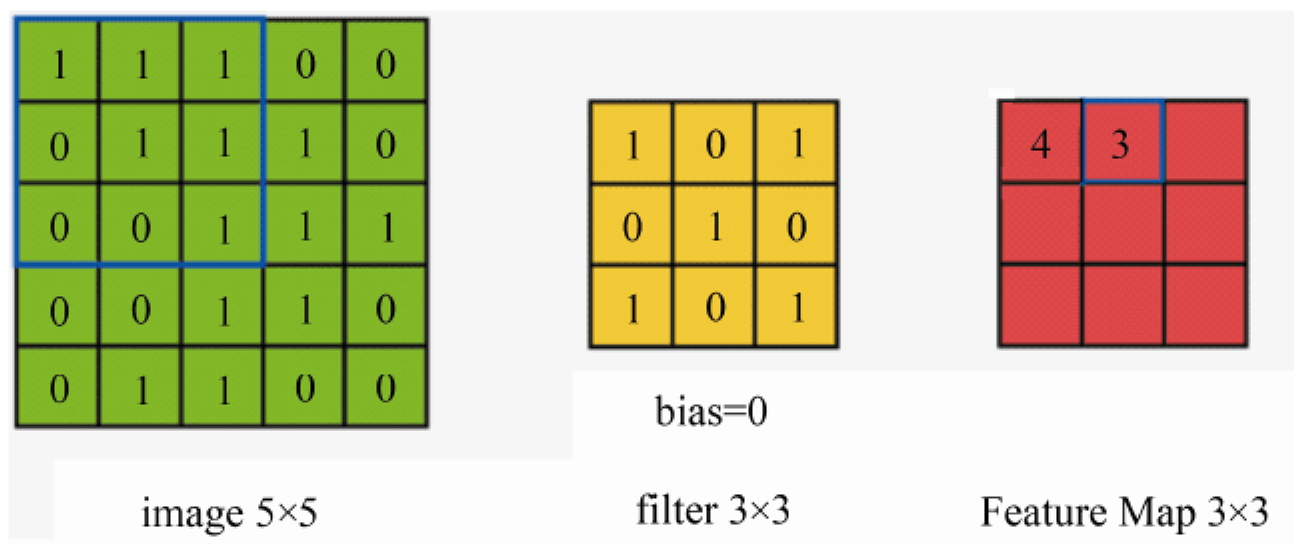


图 19-6　卷积原理图 2

同理，依次计算出 Feature Map 中所有元素的值。

在上面的计算过程中，步幅（stride）为 1，步幅可以设为大于 1 的数。例如，当步幅为 2 时 filter 将每次滑动两个元素，因此 Feature Map 就变成了 2×2。这说明图像大小、步幅和卷积后的 Feature Map 大小是有关系的，这里将不再举例。

上例仅演示了一个 filter 的情况，其实每个卷积层可以有多个 filter，每个 filter 和原始图像进行卷积后都可以得到一个 Feature Map，因此卷积后 Feature Map 的深度（个数）和卷积层的 filter 个数是相同的。图 19-7 所示为 3 个 24×24 大小的 filter（即 3×24×24）得到的三维的 Feature Map。

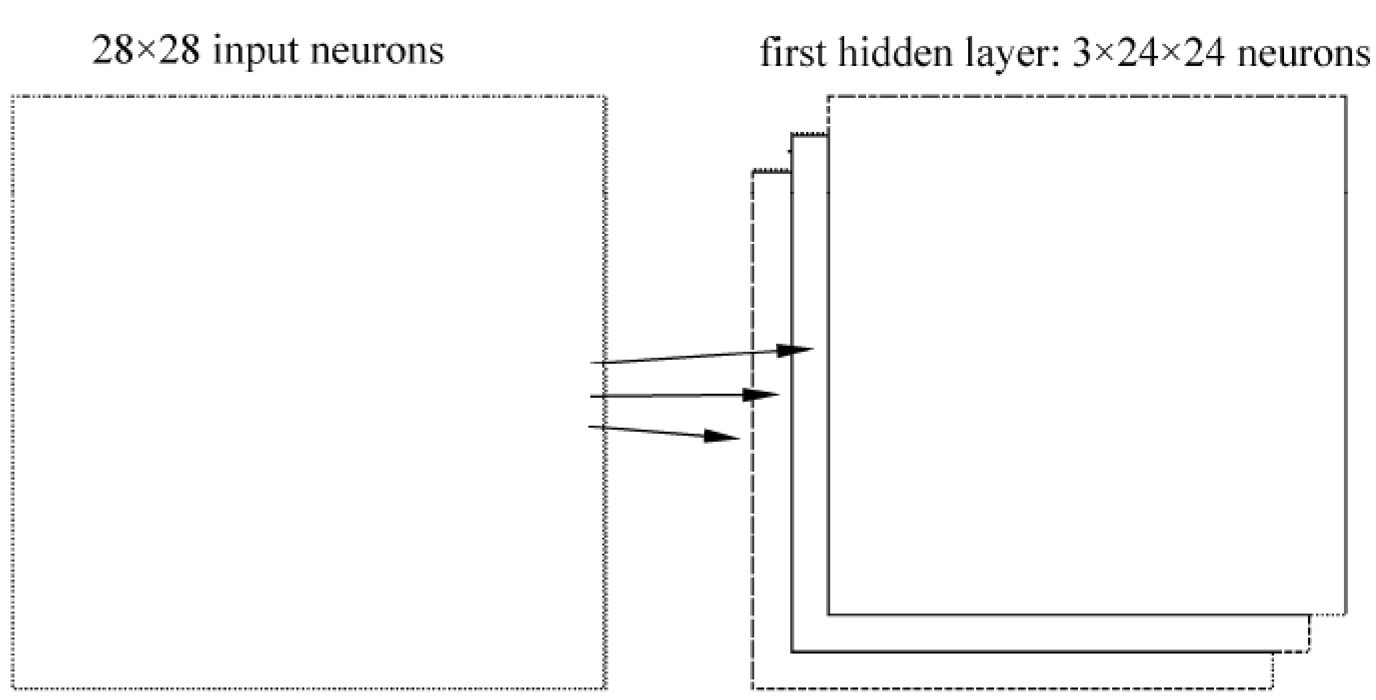


图 19-7　多个卷积核

以上就是卷积层的计算方法，这里体现了局部连接和权值共享：每层神经元只和上一层部分神经元相连（卷积计算规则），且 filter 的权值对于上一层所有神经元都是一样的。

❷ 池化（Pooling）

Pooling 层的主要作用是下采样，通过去掉 Feature Map 中不重要的样本进一步减少参数数量，且可以有效地防止过拟合。Pooling 的方法很多，最常用的是最大池化（Max Pooling）。最大池化实际上就是在 n×n 的样本中取最大值，作为采样后的样本值。

图 19-8 是 2×2 步幅为 2 的最大池化，即在获取的 Feature Map 中每 2×2 的矩阵内取最大值作为采样后的结果，这样能把数据缩小至 1/4，同时又不会损失太多信息。

对于深度为 D 的 Feature Map，各层独立做 Pooling，因此 Pooling 后的深度仍然为 D。

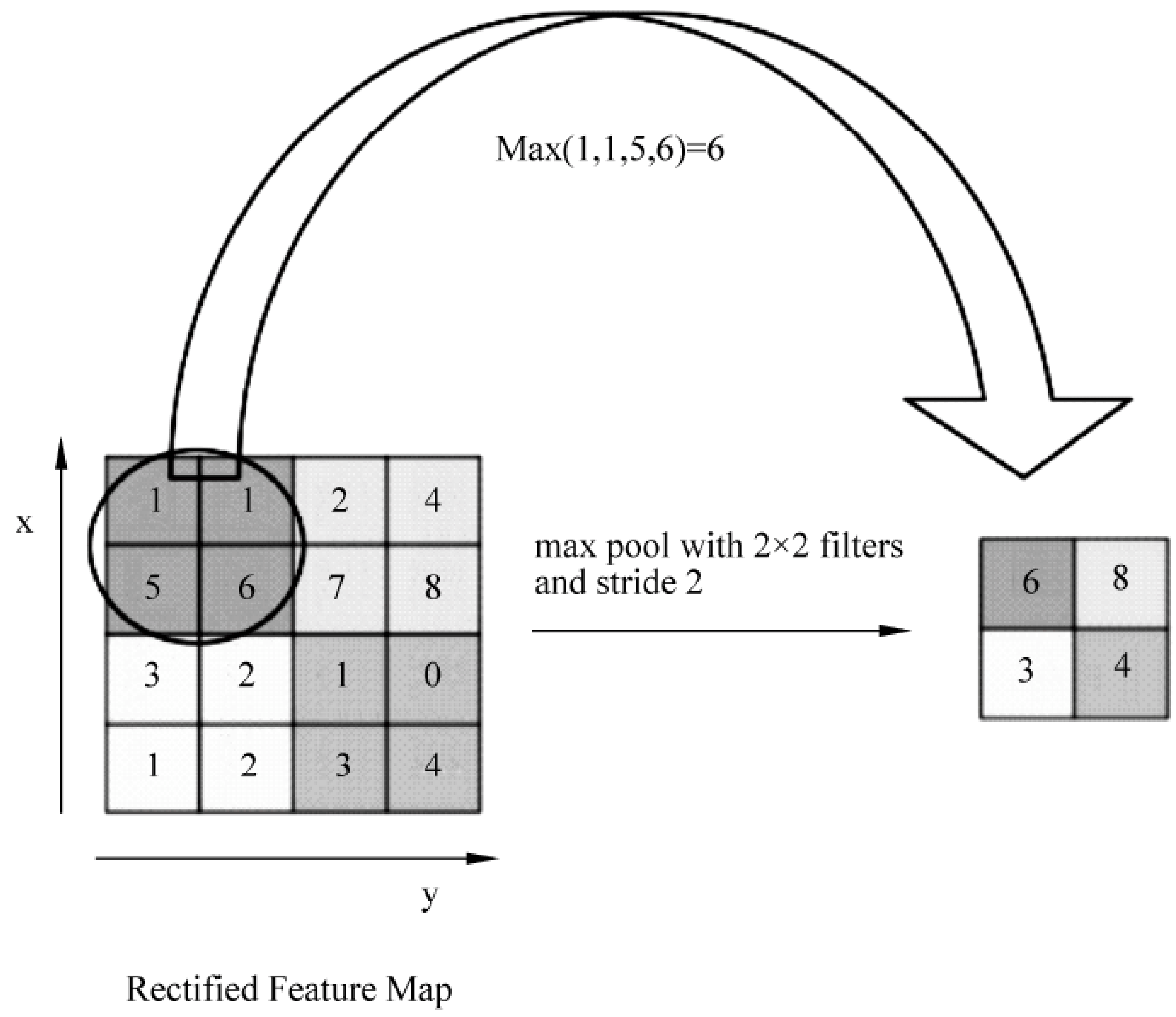


图 19-8　池化

❸ 激活函数

激活函数的作用是能够给神经网络加入一些非线性因素，使得神经网络可以更好地解决较为复杂的问题。常见的激活函数有 Sigmoid()、tanh()、ReLU()等，这里简要介绍两个常用的函数 Sigmoid()和 ReLU()。

1）Sigmoid()函数

其表达式如下：

$$g(z)=\frac{1}{1+e^{-z}}$$

其中，z 是一个线性组合，比如 z 可以等于 $w_0 + w_1 \times x_1 + w_2 \times x_2$。通过代入很大的正数或很小的负数到函数中可知，g(z)的结果趋近于 0 或 1。

因此，Sigmoid()函数的图形表示如图 19-9 所示。

也就是说，Sigmoid()函数的功能是把一个实数压缩至 0～1。当输入非常大的正数时，输出结果会接近 1；当输入非常大的负数时，则会得到接近 0 的结果。压缩至 0～1 的作用是可以把激活函数看作一种“分类的概率”，比如激活函数的输出为 0.9，便可以解释为 90%的概率为正样本。

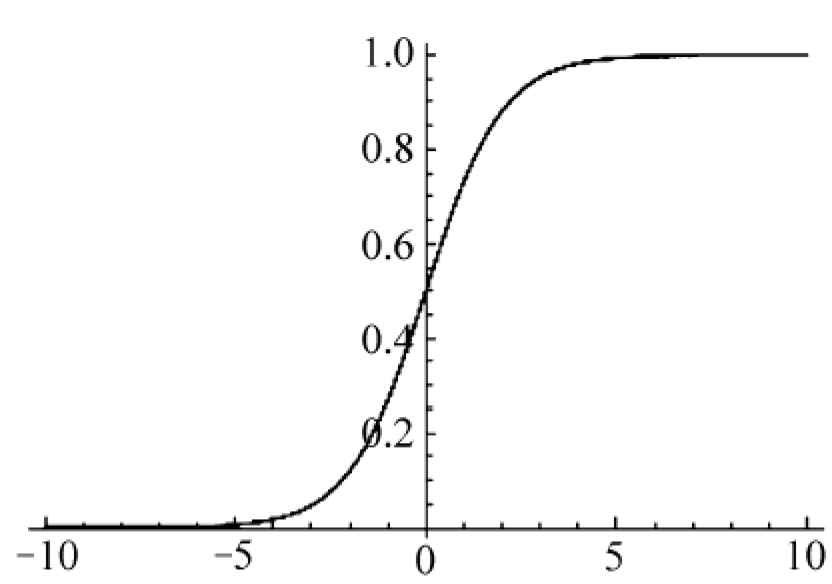


图 19-9　Sigmoid()函数图形

2）ReLU()函数

该函数被定义为：

$$y=\begin{cases}0 & (x \le 0)\\ x & (x > 0)\end{cases}$$

在 ReLU()函数中，当 x<0 时函数值为 0，否则仍为 x。ReLU()函数的图形表示如图 19-10

所示。

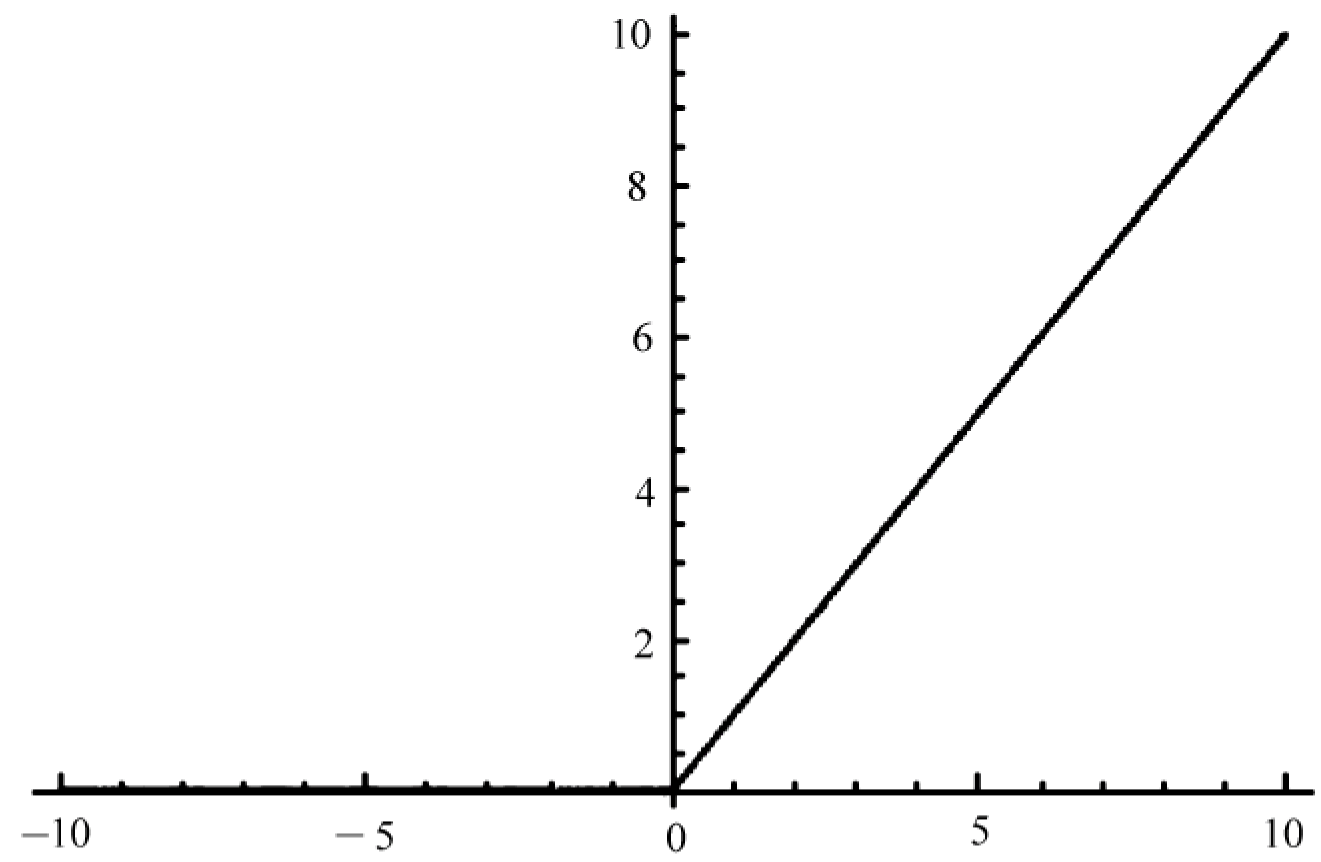


图 19-10　ReLU()函数图形

与 Sigmoid()和 ReLU()函数相比，Sigmoid()函数在输入参数太大或太小时，会产生梯度消失现象，而 ReLU()对于随机梯度下降的收敛有巨大的加速作用，且 ReLU()只需要一个阈值就可以得到激活值，而不用进行一大堆复杂的（指数）运算。但 ReLU()的缺点是，它在训练时比较脆弱，容易形成不可逆转的死亡，导致了数据多样化的丢失。

❹ 卷积神经网络的网络结构

一个卷积神经网络通常由若干卷积层、Pooling 层、全连接层组成。用户可以构建各种不同的卷积神经网络。图 19-11 所示为一个常见的卷积神经网络模型。

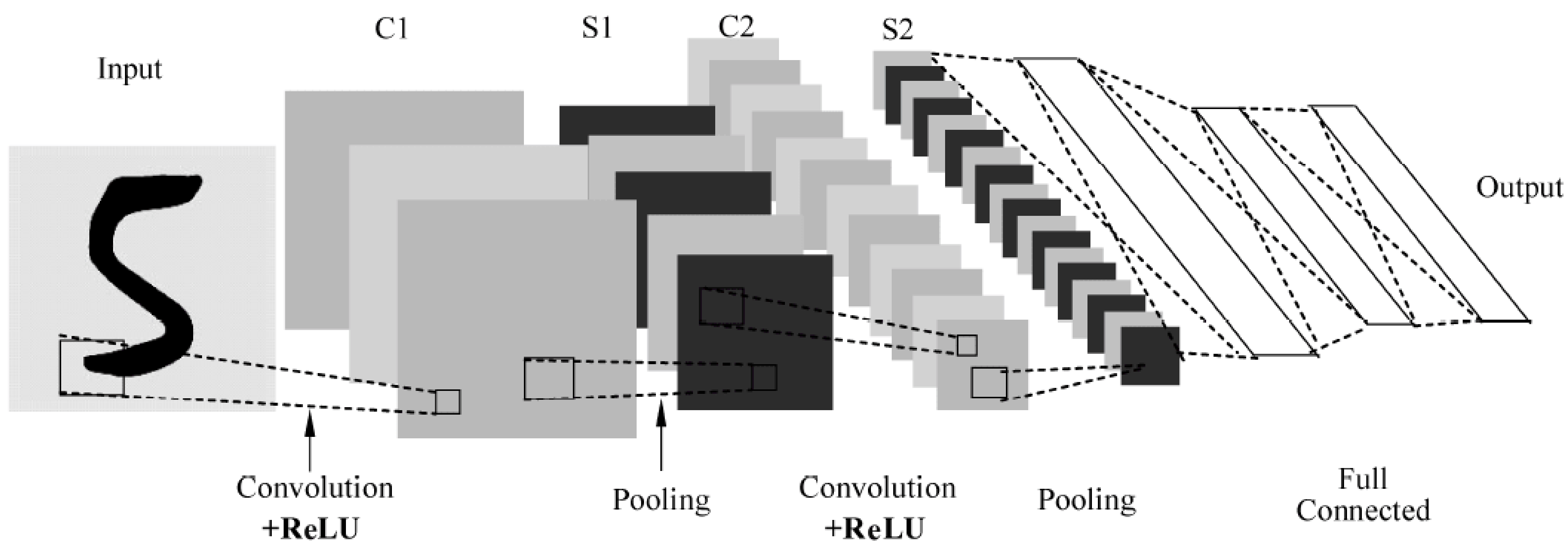


图 19-11　一个典型的 CNN 网络结构

其中，Input 为输入图像，被计算机理解为矩阵，输入图像通过 6 个可训练的 filter 进行卷积，卷积后产生 6 个特征图（Feature Map），然后再使用 ReLU()函数得到 C1 层的 Feature Map，在卷积层后面是池化（Pooling）得到 S2 层。卷积层+池化层的组合可以在隐藏层出现很多次，例如图 19-11 中出现了两次，实际上这个次数是根据模型的需要而来的。用户还可以灵活使用卷积层+卷积层或者卷积层+卷积层+池化层的组合，这些在构建模型的时候没有限制，但是最常见的 CNN 都是若干卷积层+池化层的组合，如图 19-11 中的 CNN

结构。

在若干卷积层+池化层后面是全连接层（Full Connected Layer，FC），全连接层其实就是传统的神经网络结构，即前一层的每一个神经元都与后一层的所有神经元相连，在整个卷积神经网络中起到“分类器”的作用。

19.3 Python 深度学习库——Keras

视频讲解

Keras 是搭建在 theano/tensorflow 基础上的深度学习框架，用 Python 语言编写，是一个高度模块化的神经网络库，支持 GPU 和 CPU。

19.3.1 Keras 的安装

❶ 在 Windows 下安装 Keras

在安装 Keras 之前需要安装 tensorflow、numpy、matplotlib、scipy 等库。

```
pip install numpy
pip install matplotlib
pip install scipy
pip install tensorflow
pip install keras
```

❷ 使用 Anaconda 安装 Keras

在“https://www.continuum.io/downloads/”下载对应系统版本的 Anaconda，和安装普通的软件一样，全部选择默认即可，注意勾选将 Python3.6 添加进环境变量，这样 Anaconda 就安装好了。

打开 Anaconda 菜单中的 Anaconda Prompt，输入：

```
pip install --upgrade --ignore-installed tensorflow
pip install keras
```

❸ 测试

安装完成后，在命令行下输入“python”，进入 Python 环境后输入：

```
import numpy
import scipy
import tensorflow as tf
import keras
```

若没有错误提示，则表示安装完成。

19.3.2 Keras 的网络层

Keras 的层主要包括常用层（Core）、卷积层（Convolutional）、池化层（Pooling）、局

部连接层、递归层（Recurrent）、嵌入层（Embedding）、高级激活层、规范层、噪声层、包装层，当然用户也可以编写自己的层。

对于层的操作如下：

```
layer.get_weights()                     #返回该层的权重（numpy array）
layer.set_weights(weights)              #将权重加载到该层
config=layer.get_config()               #保存该层的配置
layer=layer_from_config(config)         #加载一个配置到该层
#如果层仅有一个计算结点（即该层不是共享层），则可以通过下列方法获得输入张量、输出张量、
#输入数据的形状和输出数据的形状
layer.input
layer.output
layer.input_shape
layer.output_shape
#如果该层有多个计算结点，可以使用下面的方法
layer.get_input_at(node_index)
layer.get_output_at(node_index)
layer.get_input_shape_at(node_index)
layer.get_output_shape_at(node_index)
```

下面介绍本章案例中所使用的网络层。

❶ 二维卷积层

二维卷积层是对图像的卷积。该层对二维输入进行滑动窗卷积，当使用该层作为第 1 层时，应提供 input_shape 参数。例如 input_shape=(128,128,3)代表 128×128 的彩色 RGB 图像（data_format='channels_last'）。

操作如下：

```
keras.layers.convolutional.Conv2D(filters, kernel_size, strides=(1, 1),
padding='valid', data_format=None, dilation_rate=(1, 1), activation=None,
use_bias=True, kernel_initializer='glorot_uniform',
bias_initializer='zeros', kernel_regularizer=None, bias_regularizer=None,
activity_regularizer=None, kernel_constraint=None, bias_constraint=None)
```

参数如下。

- filters：卷积核的数目（即输出的维度）。
- kernel_size：单个整数或由两个整数构成的 list/tuple，卷积核的宽度和长度。如为单个整数，则表示在各个空间维度的相同长度。
- strides：单个整数或由两个整数构成的 list/tuple，是卷积的步长。如果为单个整数，则表示在各个空间维度的相同步长。任何不为 1 的 strides 与任何不为 1 的 dilation_rate 均不兼容。
- padding：补 0 策略，为'valid'或'same'。'valid'代表只进行有效的卷积，即对边界数据不处理。'same'代表保留边界处的卷积结果，通常会导致输出 shape 与输入 shape 相同。
- activation：激活函数，为预定义的激活函数名（参考激活函数）或逐元素

（element-wise）的 Theano 函数。如果不指定该参数，将不会使用任何激活函数（即使用线性激活函数：a(x)=x）。

- dilation_rate：单个整数或由两个整数构成的 list/tuple，指定 dilated convolution 中的膨胀比例。任何不为 1 的 dilation_rate 与任何不为 1 的 strides 均不兼容。
- data_format：字符串，'channels_first'或'channels_last'之一，代表图像的通道维的位置。该参数是 Keras 1.x 中的 image_dim_ordering，'channels_last'对应原本的'tf'，'channels_first'对应原本的'th'。以 128×128 的 RGB 图像为例，'channels_first'应将数据组织为(3,128,128)，而'channels_last'应将数据组织为(128,128,3)。该参数的默认值是~/.keras/keras.json 中设置的值，若从未设置过，则为'channels_last'。
- use_bias：布尔值，是否使用偏置项。
- kernel_initializer：权值初始化方法，为预定义初始化方法名的字符串，或用于初始化权重的初始化器。
- bias_initializer：偏置向量初始化方法，为预定义初始化方法名的字符串，或用于初始化偏置向量的初始化器。
- kernel_regularizer：施加在权重上的正则项，为 Regularizer 对象。
- bias_regularizer：施加在偏置向量上的正则项，为 Regularizer 对象。
- activity_regularizer：施加在输出上的正则项，为 Regularizer 对象。
- kernel_constraint：施加在权重上的约束项，为 Constraint 对象。
- bias_constraint：施加在偏置上的约束项，为 Constraint 对象。

❷ Dense 层（全连接层）

操作方法如下：

```
keras.layers.core.Dense(units,activation=None,use bias=True,kernel
initializer='glorot uniform',bias initializer='zeros',kernel regularizer=
None,bias regularizer=None,activity regularizer=None,kernel constraint=
None,bias_constraint=None)
```

参数如下。

- units：大于 0 的整数，代表该层的输出维度。
- use_bias：布尔值，是否使用偏置项。
- kernel_initializer：权值初始化方法，为预定义初始化方法名的字符串，或用于初始化权重的初始化器。
- bias_initializer：偏置向量初始化方法，为预定义初始化方法名的字符串，或用于初始化偏置向量的初始化器。
- regularizer：正则项，kernel 为权重的，bias 为偏执的，activity 为输出的。
- constraint：约束项，kernel 为权重的，bias 为偏执的。

❸ Activation 层

操作方法如下：

```
keras.layers.core.Activation(activation)
```

激活层对一个层的输出施加激活函数。

activation 是将要使用的激活函数，为预定义激活函数名或一个 Tensorflow/Theano 的函数。

输入 shape 任意，当使用激活层作为第 1 层时要指定 input_shape。

输出 shape 与输入 shape 相同。

❹ 最大池化层 MaxPooling2D

操作方法如下：

```
keras.layers.pooling.MaxPooling2D(pool size=(2,2),strides=None,padding=
'valid', data_format=None)
```

参数如下。

- pool_size：整数或长为 2 的整数 tuple，代表在两个方向（竖直、水平）上的下采样因子，例如取(2,2)将使图片在两个维度上均变为原长的一半。它为整数，意为各个维度值相同且为该数字。
- strides：整数或长为 2 的整数 tuple，或者为 None，步长值。
- padding：'valid'或者'same'。
- data_format：字符串，'channels_first'或'channels_last'之一，代表图像的通道维的位置。该参数是 Keras 1.x 中的 image_dim_ordering，'channels_last'对应原本的'tf'，'channels_first'对应原本的'th'。

19.3.3　用 Keras 构建神经网络

用 Keras 构建网络的过程可用图 19-12 所示。

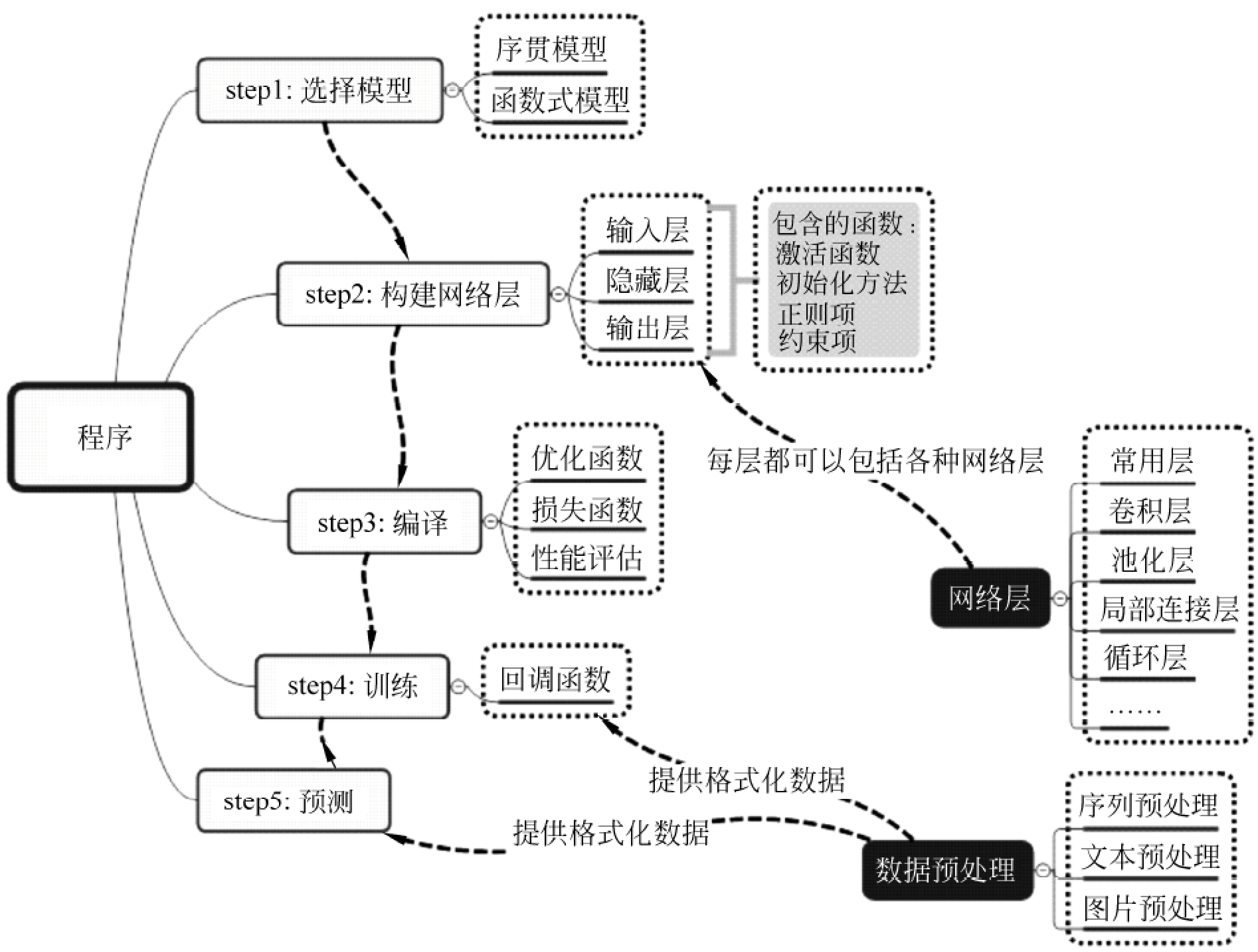


图 19-12　用 Keras 搭建神经网络

下面以一个简单的例子演示使用 Keras 如何构建网络结构并进行训练及预测。

首先引入库，并建立一个顺序模型，Sequential 就是一个空的网络结构，方法如下：

```
from keras.models import Sequential
model=Sequential()
```

在 Keras 里面可以构建一些其他的网络结构，仅需要写.add，后面加入层的类型即可。下例中引入了 Dense（也就是 fc 层）和激活函数层（RELU）：

```
from keras.layers import Dense, Activation
#再分别 add fc、relu、fc、softmax 层
model.add(Dense(units=64, input dim=100))
model.add(Activation("relu"))
model.add(Dense(units=10))
model.add(Activation("softmax"))
```

编译模型，损失函数 loss 用交叉熵，优化器用 sgd，评估用 accuracy：

```
 model.compile(loss='categorical crossentropy', optimizer='sgd', metrics=
['accuracy'])
```

载入训练数据集进行训练：

```
model.fit(x_train, y_train, epochs=5, batch_size=32)
```

对测试集进行如下操作：

```
evaluate loss_and_metrics = model.evaluate(x_test, y_test, batch_size=128)
```

19.4 程序设计的思路

视频讲解

❶ 数据集描述

在本实例中，训练样本和识别测试数据都是 28×28 像素，如图 19-13 所示的图片，它在计算机中的存储是一个二维矩阵，0 代表白色，1 代表黑色，小数代表某程度的灰色。那么输入层就应该是 28×28=786 个神经元（忽略它的二维结构），其中每个神经元的输入数据就是该像素的灰度值。整个数据集被分成两部分，即 60000 行的训练数据集和 10000 行的测试数据集，60000 行的训练数据集是一个形状为[60000,784]的张量，第 1 个维度数字用来索引图片，第 2 个维度数字用来索引每张图片中的像素点。在此张量中的每一个元素都表示某张图片中的某个像素的强度值，值的取值范围为 0～1。

输出结果只有 10 个数字（即 10 类），输出层是 10 个神经元，每个神经元对应一个要识别的结果。

❷ 网络结构

网络层级结构概述：5 层神经网络。

输入层：输入数据为原始训练图像。

第一卷积层：6 个 5×5 的卷积核，步长 Stride 为 1。在这一层中，输入为 28×28 的深度

为 1 的图片数据，输出为 24×24 的深度为 6 的特征图。

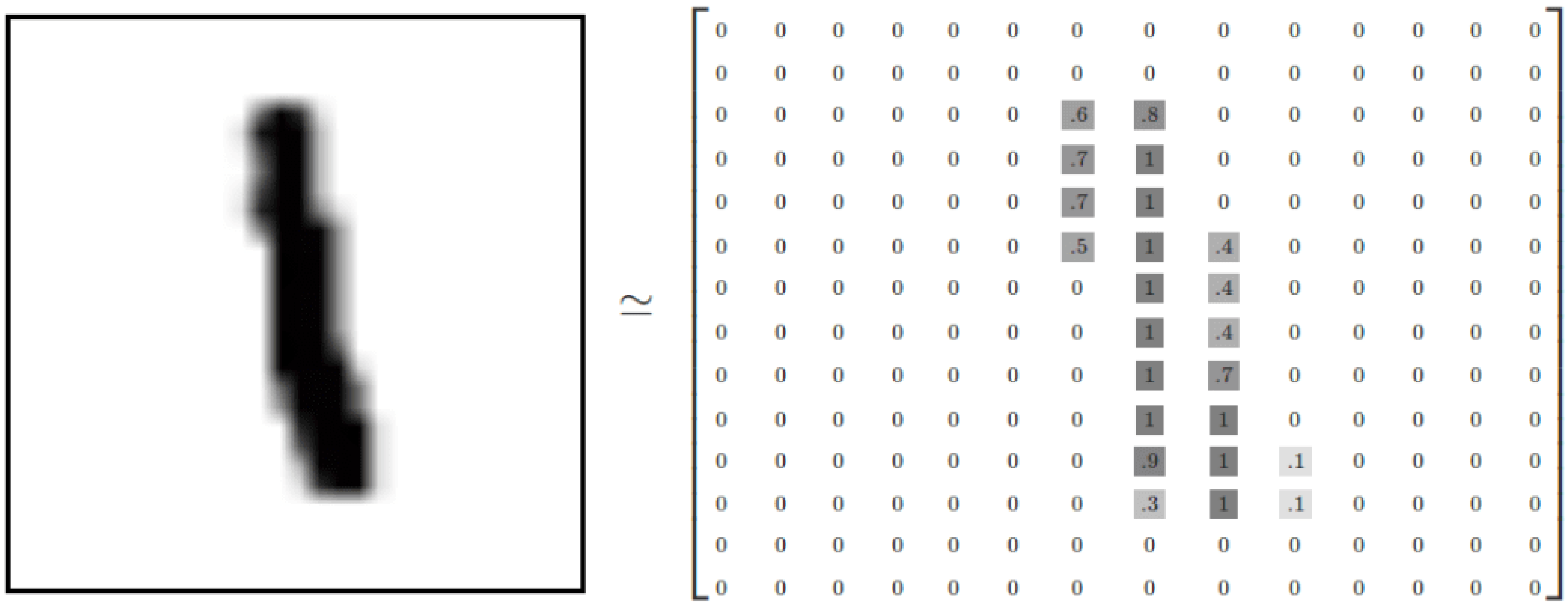


图 19-13　输入图像

第一池化层：卷积核 size 为 2×2，步长 Stride 为 2 的最大池化。在这一层中，输入为 24×24 的深度为 6 的特征图，输出为 12×12 的深度为 6 的特征图。

第二卷积层：12 个 5×5 的卷积核，步长 Stride 为 1。在这一层中，输入为 12×12 的深度为 6 的图片数据，输出为 8×8 的深度为 12 的特征图。

第二池化层：卷积核 size 为 2×2，步长 Stride 为 2 的最大池化。在这一层中，输入为 8×8 的深度为 12 的特征图，输出为 4×4 的深度为 12 的特征图。

输出层：输出为 10 维向量，激活函数为 sigmoid。

❸ 代码流程简述

（1）获取训练数据和测试数据。

（2）训练网络的超参数的定义（学习率，每次迭代中训练的样本数目，迭代次数）。

（3）构建网络层级结构。

（4）编译模型。

（5）训练模型。

（6）网络模型评估。

19.5　程序设计的步骤

19.5.1　MNIST 数据集

MNIST（Mixed National Institute of Standards and Technology database）是一个计算机视觉数据集，它包含 70000 张手写数字的灰度图片，其中每一张图片包含 28×28 个像素点。

手写体数据集（MNIST）文件的下载如下。

训练集样本：t10k-images.idx3-ubyte（下载地址为“http://luanpeng.oss-cn-qingdao.

aliyuncs.com/csdn/python/%E6%89%8B%E5%86%99%E4%BD%93%E6%95%B0%E6%8D%AE/t10k-images.idx3-ubyte”)。

训练集标签：t10k-labels.idx1-ubyte（下载地址为“http://luanpeng.oss-cn-qingdao.aliyuncs.com/csdn/python/%E6%89%8B%E5%86%99%E4%BD%93%E6%95%B0%E6%8D%AE/t10k-labels.idx1-ubyte”)。

测试集样本：train-images.idx3-ubyte（下载地址为“http://luanpeng.oss-cn-qingdao.aliyuncs.com/csdn/python/%E6%89%8B%E5%86%99%E4%BD%93%E6%95%B0%E6%8D%AE/train-images.idx3-ubyte”)。

测试集标签：train-labels.idx1-ubyte（下载地址为“http://luanpeng.oss-cn-qingdao.aliyuncs.com/csdn/python/%E6%89%8B%E5%86%99%E4%BD%93%E6%95%B0%E6%8D%AE/train-labels.idx1-ubyte”)。

19.5.2　手写体识别案例实现

❶ 读取 MNIST 数据集

编写文件 MNIST.py，用于获取手写图像数据，每张图像是 28×28 像素大小，根据需要转换成长度为 784 的行向量。

每个对象的标签为 0～9 的数字，one-hot 编码成 10 维的向量。

```
#MNIST.py 文件
import numpy as np
#数据加载器基类，派生出图片加载器和标签加载器
class Loader(object):
    #初始化加载器，path 为数据文件路径，count 为文件中的样本个数
    def __init__(self, path, count):
        self.path=path
        self.count=count
    #读取文件内容
    def get file content(self):
        print(self.path)
        f=open(self.path, 'rb')
        content=f.read()          #读取字节流
        f.close()
        return content            #字节数组

#图像数据加载器
class ImageLoader(Loader):
    #内部函数，从文件字节数组中获取第 index 个图像数据，文件中包含所有样本图片的数据
    def get picture(self, content, index):
        start=index*28*28+16
                                  #文件头 16 字节，后面每 28×28 个字节为一个图片数据
        picture=[]
        for i in range(28):
```

```
            picture.append([])                    #图片添加一行像素
            for j in range(28):
                byte1 = content[start+i*28+j]
                picture[i].append(byte1)          #在 Python3 中本来就是 int
                #picture[i].append(self.to_int(byte1))  #添加一行的每一个像素
        return picture
                    #图片为[[x,x,x…][x,x,x...][x,x,x...][x,x,x...]]的列表
    #将图像数据转化成长度为 784 的行向量形式
    def get one sample(self, picture):
        sample=[]
        for i in range(28):
            for j in range(28):
                sample.append(picture[i][j])
        return sample
    #加载数据文件，获得全部样本的输入向量，onerow 表示是否将每张图片转化为行向量，to2
    #表示是否转化为 0,1 矩阵
    def load(self,onerow=False):
        content=self.get file content()           #获取文件字节数组
        data set=[]
        for index in range(self.count):           #遍历每一个样本
            onepic=self.get picture(content, index)
                    #从样本数据集中获取第 index 个样本的图片数据，返回的是二维数组
            if onerow: onepic=self.get one sample(onepic)
                                                  #将图像转化为一维向量形式
            data set.append(onepic)
        return data set
#标签数据加载器
class LabelLoader(Loader):
    #加载数据文件，获得全部样本的标签向量
    def load(self):
        content=self.get_file_content()           #获取文件字节数组
        labels=[]
        for index in range(self.count):           #遍历每一个样本
            onelabel=content[index + 8]           #文件头有 8 个字节
            onelabelvec=self.norm(onelabel)       #one-hot 编码
            labels.append(onelabelvec)
        return labels
    #内部函数，one-hot 编码，用于将一个值转换为 10 维标签向量
    def norm(self, label):
        label vec=[]
        #label value=self.to int(label)
        label value=label  #在 Python3 中直接就是 int
        for i in range(10):
```

```
            if i==label_value:
                label vec.append(1)
            else:
                label vec.append(0)
        return label vec
#获得训练数据集，onerow 表示是否将每张图片转化为行向量
def get training data set(num,onerow=False):
    image loader=ImageLoader('train-images.idx3-ubyte', num)
                                                  #参数为文件路径和加载的样本数量
    label loader=LabelLoader('train-labels.idx1-ubyte', num)
                                                  #参数为文件路径和加载的样本数量
    return image loader.load(onerow), label loader.load()
#获得测试数据集，onerow 表示是否将每张图片转化为行向量
def get test data set(num,onerow=False):
    image loader=ImageLoader('t10k-images.idx3-ubyte', num)
                                                  #参数为文件路径和加载的样本数量
    label loader=LabelLoader('t10k-labels.idx1-ubyte', num)
                                                  #参数为文件路径和加载的样本数量
    return image_loader.load(onerow), label_loader.load()
#将一个长度为 784 的行向量打印成图形的样式
def printimg(onepic):
    onepic=onepic.reshape(28,28)
    for i in range(28):
        for j in range(28):
            if onepic[i,j]==0: print('  ',end='')
            else: print('* ',end='')
        print('')
```

❷ 训练及测试数据集

```
import numpy as np
np.random.seed(1337)  #可重现性
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Conv2D, MaxPooling2D,AveragePooling2D
import MNIST

#全局变量
batch_size=128                          #批处理样本数量
nb classes=10                           #分类数目
epochs=600                              #迭代次数
img rows, img cols=28, 28               #输入图片样本的宽、高
nb filters=32                           #卷积核的个数
pool size=(2, 2)                        #池化层的大小
kernel_size=(5, 5)                      #卷积核的大小
input_shape=(img_rows, img_cols,1)      #输入图片的维度
```

```
X train, Y train=MNIST.get training data set(6000, False)
                #加载训练样本数据集和 one-hot 编码后的样本标签数据集，最大为 60000
X test, Y test=MNIST.get test data set(1000, False)
                #加载测试特征数据集和 one-hot 编码后的测试标签数据集，最大为 10000
X train=np.array(X train).astype(bool).astype(float)/255
                                       #数据归一化
X train=X train[:,:,:,np.newaxis]
    #添加一个维度，代表图片通道，这样数据集共 4 个维度，即样本个数、宽度、高度、通道数
Y train=np.array(Y train)
X_test=np.array(X_test).astype(bool).astype(float)/255     #数据归一化
X test=X test[:,:,:,np.newaxis]
    #添加一个维度，代表图片通道，这样数据集共 4 个维度，即样本个数、宽度、高度、通道数
Y test=np.array(Y test)
print('样本数据集的维度：', X_train.shape,Y_train.shape)
print('测试数据集的维度：', X test.shape,Y test.shape)
#构建模型
model=Sequential()
model.add(Conv2D(6,kernel size,input shape=input shape,strides=1))
                                                           #卷积层 1
model.add(AveragePooling2D(pool_size=pool_size,strides=2))    #池化层
model.add(Conv2D(12,kernel_size,strides=1))                    #卷积层 2
model.add(AveragePooling2D(pool_size=pool_size,strides=2))    #池化层
model.add(Flatten())                                        #拉成一维数据
model.add(Dense(nb_classes))                                #全连接层 2
model.add(Activation('sigmoid'))                            #sigmoid 评分
#编译模型
model.compile(loss='categorical crossentropy',optimizer='adadelta',
metrics=['accuracy'])
#训练模型
model.fit(X train, Y train, batch size=batch size, epochs=epochs,
verbose=1, validation data=(X test, Y test))
#评估模型
score=model.evaluate(X test, Y test, verbose=0)
print('Test score:', score[0])
print('Test accuracy:', score[1])
#保存模型
model.save('cnn_model.h5')   #HDF5 文件，pip install h5py
```

输出结果如下：

```
Test score: 0.18881544216349722
Test accuracy: 0.959
```

由于训练时间太久，读者可自行减少训练集及测试集的数量，或者迭代次数，观察训练结果。

19.5.3 预测自己手写图像

❶ 制作自己的手写图像

使用画图工具或 Photoshop 制作一个 28×28 像素的黑底白字的图像文件，如图 19-14 所示。

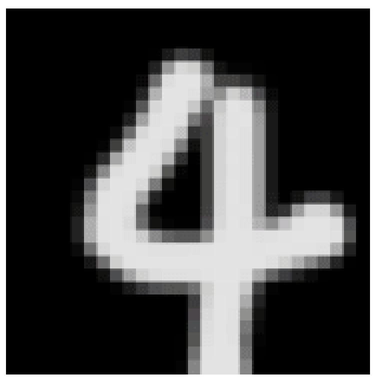


图 19-14 自己的手写图像

❷ 编写代码

新建文件 my_predict.py，编写代码如下：

```
from keras.models import load model
import numpy as np
import cv2
model=load model('cnn model.h5')
image=cv2.imread('4.png', 0)
img=cv2.imread('4.png', 0)
img=np.reshape(img,(1,28,28,1)).astype(bool).astype("float32")/255
my proba=model.predict proba(img)
my predict=model.predict classes(img)
print('识别为：')
print(my proba)
print(my predict)
cv2.imshow("Image1", image)
cv2.waitKey(0)
```

第 20 章

词云实战——爬取豆瓣影评生成词云

20.1 功能介绍

视频讲解

“词云”就是对网络文本中出现频率较高的“关键词”予以视觉上的突出，形成“关键词云层”或“关键词渲染”，从而过滤掉大量的文本信息，使浏览网页者只要一眼扫过文本就可以领略文本的主旨。

豆瓣电影提供了最新的电影介绍及评论，包括上映影片的影讯查询及购票服务，观众可以记录想看、在看和看过的电影/电视剧，以及打分、写影评。豆瓣电影会根据观众的口味推荐好电影。本程序使用 Python 爬虫技术获取豆瓣电影（https://movie.douban.com/）中最新电影的影评，经过数据清理和词频统计后对电影《黑豹》的影评信息进行词云展示，效果如图 20-1 所示。

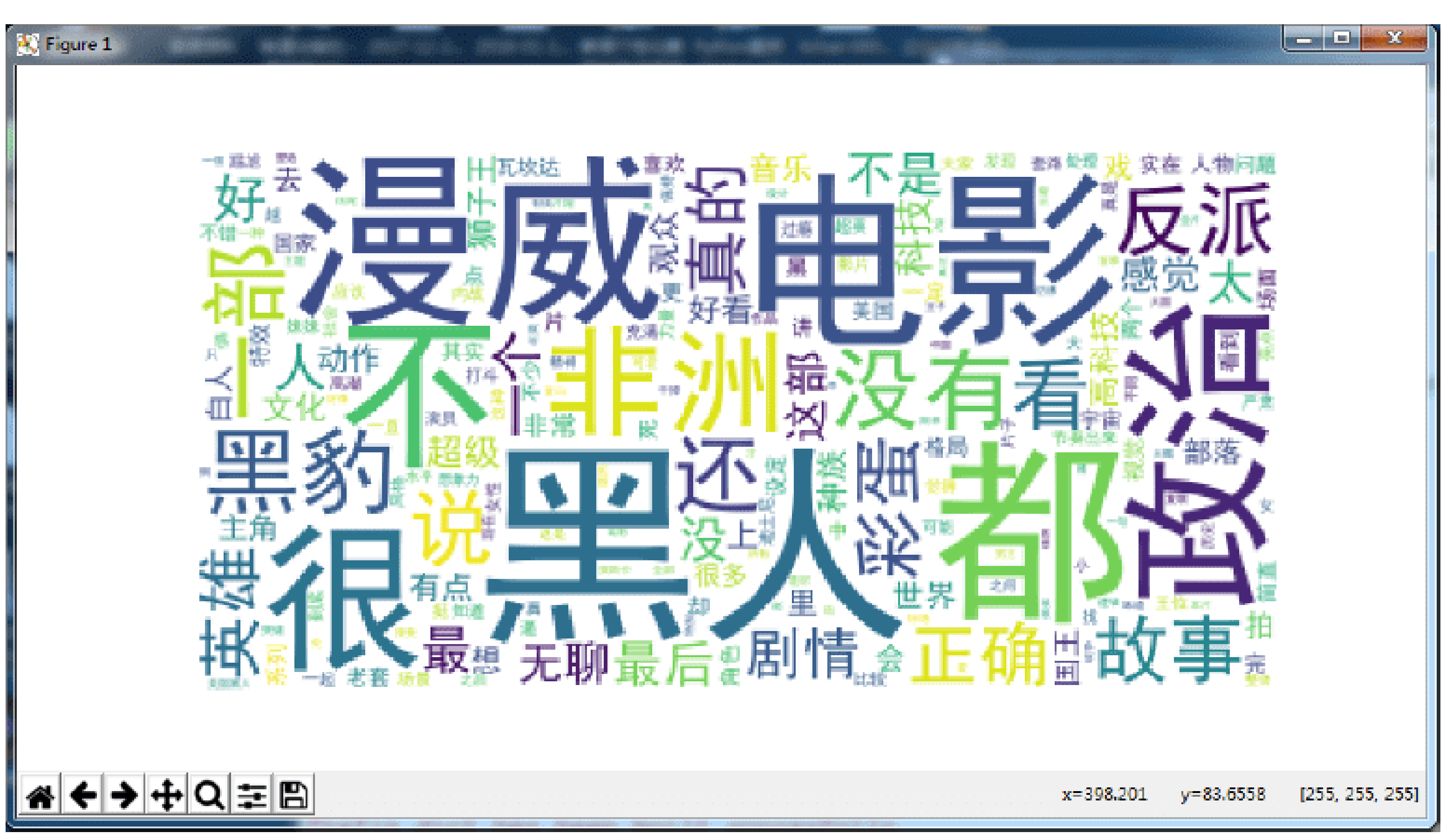


图 20-1 《黑豹》影评信息的词云显示结果

20.2 程序设计的思路

本程序主要分为 3 个过程。

❶ 抓取网页数据

使用 Python 爬虫技术获取豆瓣电影中最新上映电影的网页（如图 20-2 所示），其网址如下：

https://movie.douban.com/cinema/nowplaying/zhengzhou/

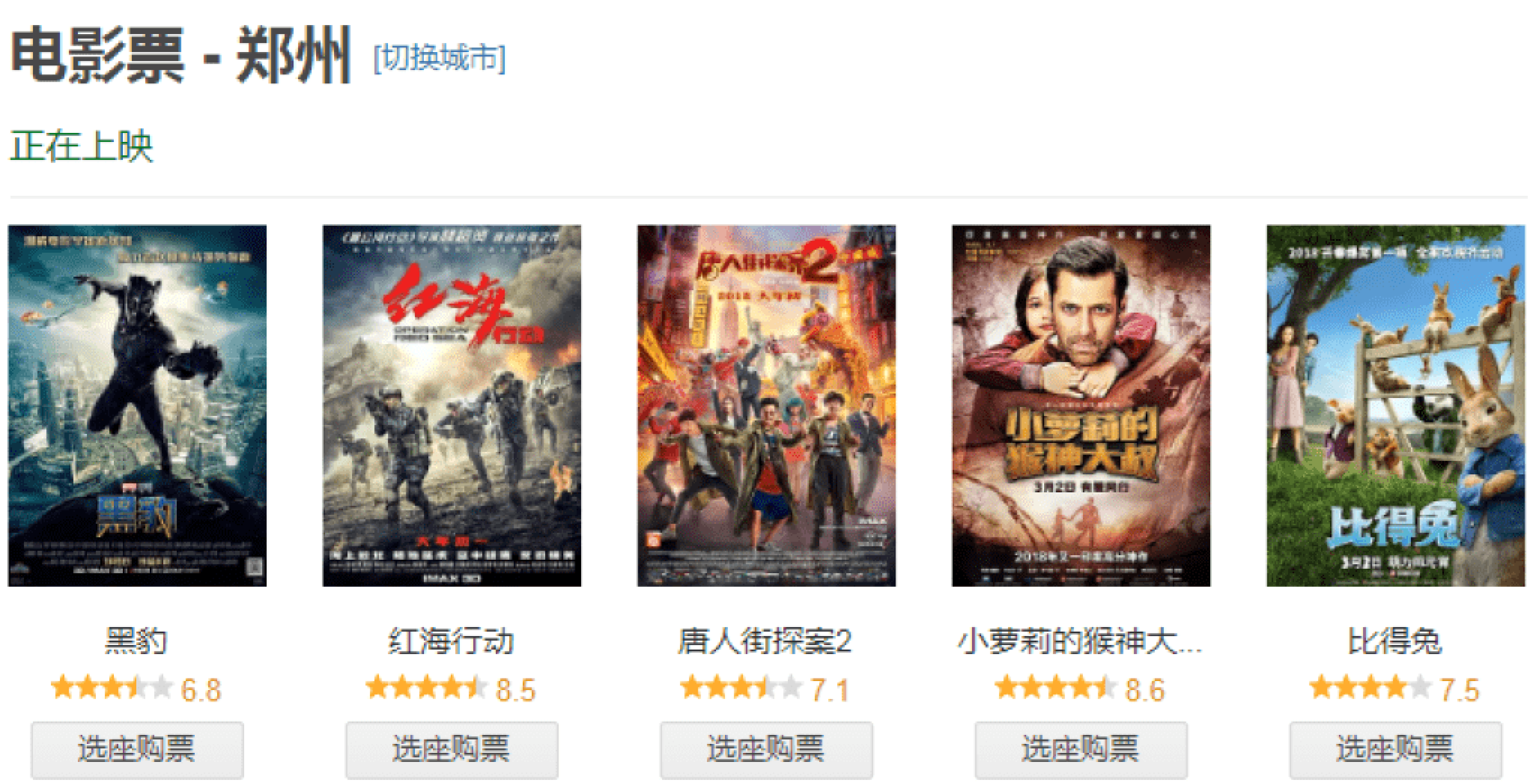


图 20-2　最新上映电影的网页

通过其 HTML 解析出每部电影的 ID 号和电影名，获取某 ID 号就可以得到该部电影的影评网址，形式如下：

https://movie.douban.com/subject/26861685/comments

其中，26861685 就是电影《红海行动》的 ID 号。这样仅仅获取了 20 个影评，可以指定开始号 start 来获取更多影评，例如：

https://movie.douban.com/subject/26861685/comments?start=40&limit=20

这意味着获取从第 40 条开始的 20 个影评。

❷ 清理数据

通常将某部影评信息存入 eachCommentList 列表中。为便于数据清理和词频统计，把 eachCommentList 列表形成字符串 comments，将 comments 字符串中的“也”“太”“的”等虚词（停用词）清理掉后进行词频统计。

❸ 用词云进行展示

最后使用词云包对影评信息进行词云展示。

20.3　关键技术

视频讲解

20.3.1　安装 WordCloud

WordCloud 使用最常规的 pip install wordcloud 命令安装。

如果安装失败，可以使用 Windows 二进制安装包（WHL 文件）直接安装，步骤如下：

首先转到“http://www.lfd.uci.edu/~gohlke/pythonlibs/#wordcloud”页面，下载需要的对应版本的 WordCloud 的 WHL 文件。如果用户使用的是 64bit 的 Python3.5，请下载图 20-3 中框住的文件。

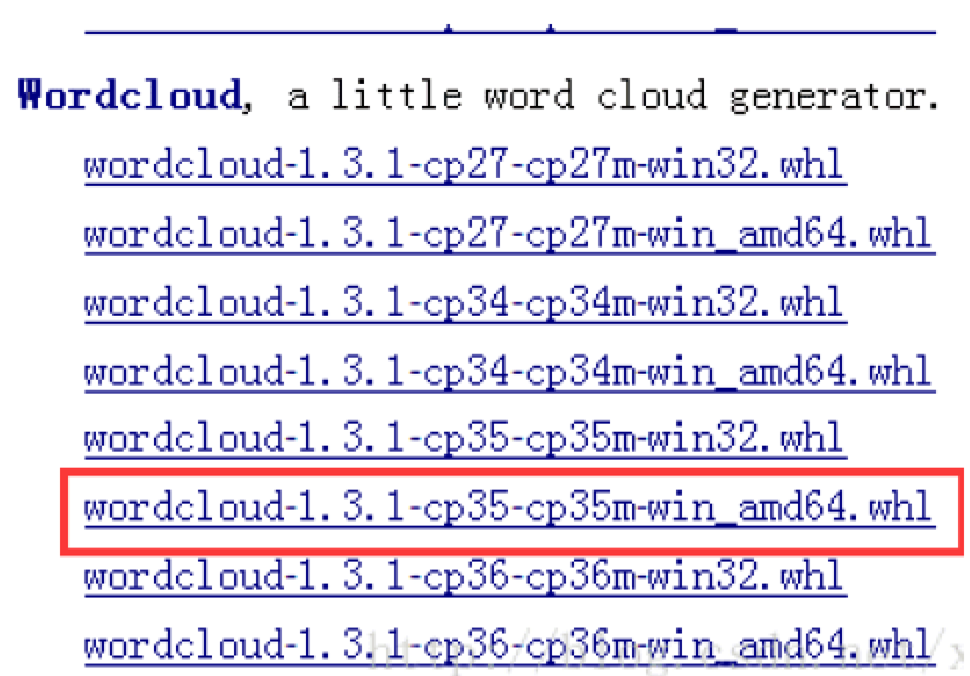


图 20-3　下载文件

然后在 cmd 命令行中进入到刚刚下载的文件的路径，使用 pip install wordcloud-1.3.1-cp35-cp35m-win_amd64.whl 命令开始安装，大约一分钟就可以安装完成。

20.3.2　使用 WordCloud

❶ WordCloud 的基本用法

```
class wordcloud.WordCloud(font_path=None, width=400, height=200, margin=2,
ranks_only=None, prefer_horizontal=0.9, mask=None, scale=1,
color_func=None, max_words=200, min_font_size=4, stopwords=None,
random_state=None, background_color='black', max_font_size=None,
font_step=1, mode='RGB', relative_scaling=0.5, regexp=None,
collocations=True, colormap=None, normalize_plurals=True)
```

这是 WordCloud 的所有参数，下面具体介绍一下各参数。

- font_path：需要展现什么字体就把该字体路径+扩展名写上，例如 font_path = '黑体.ttf'。
- width：输出的画布宽度，默认为 400 像素。

- height：输出的画布高度，默认为 200 像素。
- prefer_horizontal：词语水平方向排版出现的频率，默认为 0.9（所以词语垂直方向排版出现的频率为 0.1）。
- mask：如果该参数为空，则使用二维遮罩绘制词云；如果该参数非空，设置的宽/高值将被忽略，遮罩形状将被 mask 取代。除了全白（#FFFFFF）部分不会绘制以外，其余部分会用于绘制词云。例如 bg_pic = imread('读取一张图片.png')，背景图片的画布一定要设置为白色（#FFFFFF），然后显示的形状为不是白色的其他颜色。用户可以用 PS 工具将自己要显示的形状复制到一个纯白色的画布上，然后保存。
- Scale：按照比例放大画布，例如设置为 1.5，则长和宽都是原来画布的 1.5 倍。
- min_font_size：显示的最小的字体大小。
- font_step：字体步长，如果步长大于 1，会加快运算，但是可能导致结果出现较大的误差。
- max_words：要显示的词的最大个数。
- stopwords：设置需要屏蔽的词，如果为空，则使用内置的 STOPWORDS。
- background_color：背景颜色，例如 background_color='white'，背景颜色为白色，默认颜色为黑色。
- max_font_size：显示的最大的字体大小。
- mode：当该参数为“RGBA”并且 background_color 不为空时背景透明。
- relative_scaling：词频和字体大小的关联性。
- color_func：生成新颜色的函数，如果为空，则使用 self.color_func。
- regexp：使用正则表达式分隔输入的文本。
- collocations：是否包括两个词的搭配。
- colormap：给每个单词随机分配颜色，若指定 color_func，则忽略该方法。

WordCloud 提供的方法如下。

- fit_words(frequencies)：根据词频生成词云。
- generate(text)：根据文本生成词云。
- generate_from_frequencies(frequencies[,...])：根据词频生成词云。
- generate_from_text(text)：根据文本生成词云。
- process_text(text)：将长文本分词并去除屏蔽词（此处指英语，中文分词还需要自己用其他库先行实现，使用上面的 fit_words(frequencies)）。
- recolor([random_state,color_func,colormap])：对现有输出重新着色，重新着色会比重新生成整个词云快很多。
- to_array()：转化为 numpy array。
- to_file(filename)：输出到文件。

❷ WordCloud 的应用举例

```
#导入 wordcloud 模块和 matplotlib 模块
from wordcloud import WordCloud,ImageColorGenerator,STOPWORDS
import matplotlib.pyplot as plt
from scipy.misc import imread
```

```
text=open('test.txt','r').read()          #读取一个 TXT 文件
bg_pic=imread('alice.png')                #读入背景图片
'''设置词云样式'''
wc=WordCloud(
    background_color='white',
                       #background color 参数用于设置背景颜色，默认颜色为黑色
    mask=bg pic,
    #有中文这句代码必须添加，否则会只出现方框而不出现汉字
    font path='simhei.ttf',               #通过 font path 参数来设置字体集
    max words=2000,
    max font size=150,
    random state=30,scale=1.5)
wc.generate_from_text(text)               #根据文本生成词云
image colors=ImageColorGenerator(bg pic)
plt.imshow(wc)                            #显示词云图片
plt.axis('off')
plt.show()
print('display success!')
wc.to_file('test2.jpg')                   #保存图片
```

只有在设置 mask 的情况下才会得到一个拥有图片形状的词云。本程序使用的模板图是 alice.png（如图 20-4 所示），生成的词云形状如图 20-5 所示。



图 20-4　模板图



图 20-5　生成的词云图

❸ 设置停用词

用户也可以设置停用词（“太”“的”等虚词），使得词云中不显示该虚词，例如：

```
from os import path
from PIL import Image
```

```
import numpy as np
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
#读取整个文章
text=open('test.txt','r').read()        #读取一个 TXT 文件
#读取遮罩/彩色图像
alice coloring=np.array(Image.open(path.join(d, "alice color.png")))
#设置停用词
stopwords=set(STOPWORDS)
stopwords.add("的")               #人工添加停用词
stopwords.add("了")               #人工添加停用词
#可以通过 mask 参数来设置词云形状
wc=WordCloud(background color="white", max words=2000, mask=alice
coloring,stopwords=stopwords, max font size=40, random state=42)
#生成词云
wc.generate(text)
#根据图片生成颜色
image colors=ImageColorGenerator(alice coloring)
plt.imshow(wc, interpolation="bilinear")
plt.axis("off")
plt.show()
```

❹ WordCloud 使用词频

```
import jieba.analyse
from PIL import Image,ImageSequence
import numpy as np
import matplotlib.pyplot as plt
from wordcloud import WordCloud,ImageColorGenerator
lyric=''
f=open('./test.txt','r')
for i in f:
    lyric+=f.read()
#用 jieba 对文章做分词，提取出词频高的前 50 个词
result=jieba.analyse.textrank(lyric,topK=50,withWeight=True)
keywords=dict()
for i in result:
    keywords[i[0]]=i[1]
print(keywords)
```

输出如下：

```
{'听见':0.26819943990969086,'风雨':0.4369472045572426,'不能':
0.41455711258992367,'生命':0.5934235845918548,'理想':0.26510877668765925,'星河':
0.23975886019089002,'青春':0.24010255181105905,'希望':0.494835681401572,'痛算':
```

```
0.27939232288036236,'梦想':0.6607157592927637,'大家':0.2630206522238169,'日落':
0.24124829031114808,'相信':1.0,'随风':0.24679509210701486,'热血':
0.3747213789071734,'怒放':0.4236733731776506,'忘掉':0.37456984152879724,'卷起':
0.28349442481975,'兄弟':0.41239452275709515,'超越':0.39647241012049056,'英雄':
0.31311037526655513,'像是':0.30426828861337796,'跌倒':0.3625975500392993,'想要':
0.5829550209468557,'命运':0.7201128992940313,'变化':0.2686953866879604,'天空':
0.3146976061015469,'父亲':0.24636152229739733,'世界':0.3565812143701714,'没有':
0.5977870162380065,'人生':0.3775236250279759,'生活':0.2663673685783774,'改变':
0.8023053505916324,'穿行':0.30336139077497054,'海洋':0.28687650921373503,'追逐':
0.2816469457707918 6,'拥有':0.551167695718 6838,'太阳':0.3128100115945511 3,'知道':
0.28305393123835487,'拍拍':0.2877289851675474,'摇摆':0.4813790823694424,'力量':
0.5692829648461694,'翅膀':0.36632797341678375,'朋友':0.2528034375864833,'挣脱':
0.39383738344839236,'奔跑':0.4640807450464461,'方向':0.4093246167577443,'就算':
0.9832790417761437,'水手':0.3471435439240663,'忘记':0.23862724809926258}
    image=Image.open('./tim.png')
    graph=np.array(image)
    wc=WordCloud(font path='./fonts/simhei.ttf',background color='White',
    max words=50,mask=graph)
    wc.generate_from_frequencies(keywords)        #词频生成词云
    image color=ImageColorGenerator(graph)
    plt.imshow(wc)
    plt.imshow(wc.recolor(color func=image color))
    plt.axis("off")
    plt.show()
    wc.to_file('dream.png')
```

20.4　程序设计的步骤

视频讲解

❶ 抓取网页数据

首先要对网页进行访问，在 Python 中使用的是 urllib 库，代码如下：

```
from urllib import request
resp=request.urlopen('https://movie.douban.com/nowplaying/hangzhou/')
html_data=resp.read().decode('utf-8')
```

其中，“https://movie.douban.com/cinema/nowplaying/zhengzhou/”是豆瓣电影最新上映的电影页面，用户可以在浏览器中输入该网址进行查看；html_data 是字符串类型的变量，里面存放了网页的 HTML 代码。输入 print(html_data)可以查看最新上映影片的影讯信息，如图 20-6 所示。

然后对得到的 HTML 代码进行解析，提取出自己需要的数据。在 Python 中使用 BeautifulSoup 库（如果没有则使用 pip install BeautifulSoup 进行安装）进行 HTML 代码的解析。

```
<div id="nowplaying">
    <div class="mod-hd">
        <h2>正在上映</h2>
    </div>
    <div class="mod-bd">
        <ul class="lists">
                <li
                    id="6390825"
                    class="list-item"
                    data-title="黑豹"
                    data-score="6.8"
                    data-star="35"
                    data-release="2018"
                    data-duration="135分钟(中国大陆)"
                    data-region="美国"
                    data-director="瑞恩·库格勒"
                    data-actors="查德维克·博斯曼 / 露皮塔·尼永奥 / 迈克尔·B·乔丹"
                    data-category="nowplaying"
                    data-enough="True"
                    data-showed="True"
                    data-votecount="64156"
                    data-subject="6390825"
                >
```

图 20-6　最新上映影片的影讯信息

BeautifulSoup 使用的格式如下：

```
BeautifulSoup(html,"html.parser")
```

第 1 个参数为需要提取数据的 HTML，第 2 个参数是指定解析器，然后使用 find_all() 读取 HTML 中的内容。

但是 HTML 中有那么多的标签，该读取哪些呢？其实，最简单的办法是打开爬取网页的 HTML 代码，然后查看需要的数据在哪个 HTML 标签里面，如图 20-6 所示。

由图 20-6 可以看出，从<div id='nowplaying'>标签开始是想要的数据，里面有电影的名称、评分、主演等信息，所以相应的代码编写如下：

```
from bs4 import BeautifulSoup as bs
soup=bs(html data, 'html.parser')
nowplaying movie=soup.find all('div', id='nowplaying')
nowplaying movie list=nowplaying movie[0].find all('li', class =
'list-item')
```

其中，nowplaying_movie_list 是所有电影信息的一个列表，可以用 print(nowplaying_movie_list[1])查看第 2 部影片《红海行动》的内容，如图 20-7 所示。

从该图中可以看到在 data-subject 属性里面放了电影的 ID 号，而在 img 标签的 alt 属性里面放了电影的名字，因此通过这两个属性来得到电影的 ID 和名称（在打开电影短评的网页时需要用到电影的 ID，所以需要对它进行解析），编写代码如下：

```
nowplaying list=[]
for item in nowplaying movie list:
        nowplaying_dict={}        #以字典形式存储每部电影的 ID 和名称
```

```
        nowplaying dict['id']=item['data-subject']
        for tag_img_item in item.find_all('img'):
            nowplaying dict['name']=tag img item['alt']
            nowplaying_list.append(nowplaying_dict)
```

```
<li
    id="26861685"
    class="list-item"
    data-title="红海行动"
    data-score="8.5"
    data-star="45"
    data-release="2018"
    data-duration="138分钟"
    data-region="中国大陆 香港"
    data-director="林超贤"
    data-actors="张译 / 黄景瑜 / 海清"
    data-category="nowplaying"
    data-enough="True"
    data-showed="True"
    data-votecount="312987"
    data-subject="26861685"
>
    <ul class="">
        <li class="poster">
            <a href="https://movie.douban.com/subject/26861685/?from=playing_poster" class=ticket-btn target="_blank" data-psource="poster">
                <img src="https://img3.doubanio.com/view/photo/s_ratio_poster/public/p2514119443.webp" alt="红海行动" rel="nofollow" class="" />
            </a>
        </li>
        <li class="stitle">
            <a href="https://movie.douban.com/subject/26861685/?from=playing_poster"
                class="ticket-btn"
                target="_blank"
                title="红海行动"
                data-psource="title">
                红海行动
            </a>
        </li>
```

图 20-7　《红海行动》电影信息的 HTML 标签

在列表 nowplaying_list 中存放了最新电影的 ID 和名称，可以使用 print(nowplaying_list) 进行查看，结果如下：

```
[{'id': '6390825', 'name': '黑豹'}, {'id': '26861685', 'name': '红海行动'},
{'id': '26698897', 'name': '唐人街探案 2'}, {'id': '26393561', 'name': '
小萝莉的猴神大叔'}, {'id': '26649604', 'name': '比得兔'}, {'id': '26603666',
'name': '妈妈咪鸭'}, {'id': '30152451', 'name': '厉害了，我的国'}, {'id':
'26972275', 'name': '恋爱回旋'}, {'id': '26575103', 'name': '捉妖记 2'},
{'id': '27176717', 'name': '熊出没·变形记'}, {'id': '26611804', 'name':
'三块广告牌'}, {'id': '25829175', 'name': '西游记女儿国'}, {'id': '27085923',
'name': '灵魂当铺之时间典当'}, {'id': '27114417', 'name': '祖宗十九代'},
{'id': '25899334', 'name': '飞鸟历险记'}, {'id': '3036465', 'name':
'爱在记忆消逝前'}, {'id': '27180882', 'name': '疯狂的公牛'}, {'id': '25856453',
'name': '闺蜜 2'}, {'id': '26836837', 'name': '宇宙有爱浪漫同游'}]
```

可以看到是和豆瓣网址上面匹配的，这样就得到了最新电影的信息。接下来对最新电影短评进行分析。例如《红海行动》的短评网址为“https://movie.douban.com/subject/26861685/comments?start=0&limit=20”，其中 26861685 就是《红海行动》电影的 ID，start=0 表示第 0 条评论。

查看上面的短评页面的 HTML 代码，可以发现关于《红海行动》评论的数据在 div 标签的 comment 属性下面，如图 20-8 所示。

```
<div class="comment">
    <h3>
        <span class="comment-vote">
            <span class="votes">8801</span>
            <input value="1323425806" type="hidden"/>
            <a href="javascript:;" class="j a_show_login" onclick="">有用</a>
        </span>
        <span class="comment-info">
            <a href="https://www.douban.com/people/dreamfox/" class="">乌鸦火堂</a>
                <span>看过</span>
                <span class="allstar40 rating" title="推荐"></span>
            <span class="comment-time " title="2018-02-13 15:35:16">
                2018-02-13
            </span>
        </span>
    </h3>
    <p class=""> 春节档最好！最好不是战狼而是战争，有点类似黑鹰坠落，主旋律色彩下，真实又残酷的战争渲染。故事性不强，文戏不超20分钟，从头打到尾，林超贤场面调度极佳，巷战、偷袭、突击有条不紊，军械武器展示效果不错。尺度超大，钢锯岭式血肉横飞，还给你看特写！敌人如丧尸一般打不完，双方的狙击手都是亮点
    </p>
</div>
```

图 20-8 《红海行动》短评信息的 HTML 标签

因此对该标签进行解析，代码如下：

```
requrl='https://movie.douban.com/subject/' + nowplaying_list[0]['id'] +
'/comments' +'?' +'start=0' + '&limit=20'
resp=request.urlopen(requrl)
html_data=resp.read().decode('utf-8')
soup=bs(html_data, 'html.parser')
comment_div_lits=soup.find_all('div', class_='comment')
```

此时在 comment_div_lits 列表中存放的就是 class_='comment'的所有 div 标签里面的 HTML 代码了。在图 20-8 中还可以发现<div class_='comment'>标签里面的 p 标签下面存放了网友对电影的评论，因此对 comment_div_lits 代码中的 HTML 代码继续进行解析，代码如下：

```
eachCommentList=[];
for item in comment_div_lits:
       if item.find_all('p')[0].string is not None:
           eachCommentList.append(item.find_all('p')[0].string)
```

使用 print(eachCommentList)查看 eachCommentList 列表中的内容，可以看到里面存放了大家想要的影评。

至此已经爬取了豆瓣电影最近播放电影的评论数据，接下来就要对数据进行清洗和词云显示了。

❷ 数据清洗

数据清洗是消去与数据分析无关的信息，这里为了方便进行数据清洗，将列表中的数据放在一个字符串中，代码如下：

```
comments=''
for k in range(len(eachCommentList)):
comments=comments + (str(eachCommentList[k])).strip()
```

使用 print(comments)进行查看，可以看到所有的评论已经变成一个字符串，但是评论中还有很多标点符号等。这些符号对词频统计根本没用，因此要将它们清除，所用的方法是使用正则表达式，Python 中的正则表达式是通过 re 模块实现的。其代码如下：

```
import re
pattern=re.compile(r'[\u4e00-\u9fa5]+')
filterdata=re.findall(pattern, comments)
cleaned_comments=''.join(filterdata)
```

继续使用 print（cleaned_comments）语句进行查看，可以看到此时评论数据中已经没有那些标点符号了，数据被清“干净”了。

因为要进行词频统计，所以先进行中文分词操作。在这里使用的是 jieba 分词，如果用户没有安装 jieba 分词，可以在控制台使用 pip install jieba 进行安装（可以使用 pip list 查看是否安装了这些库）。中文分词的代码如下：

```
import jieba.analyse                        #分词包
#使用 jieba 分词进行中文分词
result=jieba.analyse.textrank(cleaned_comments,topK=50,withWeight=True)
keywords=dict()
for i in result:
    keywords[i[0]]=i[1]
print("删除停用词前",keywords)
```

结果如下：

```
{'大片': 0.28764823530539835, '动作': 0.42333889433557714, '人质':
0.18041389646505365, '不能': 0.18403284248005652, '行动':
0.5258110409848542, '的': 0.19000741337241692, '看到': 0.16410604936619055,
'太': 0.250308587701313, '导演': 0.3247672024874971, '军人':
0.22827987403008904, '主旋律': 0.21300534948534544, '电影': 1.0, '作战':
0.17912699218043704, '震撼': 0.24439586499277743, '国产':
0.37209344994813087, '人物': 0.3211015399811397, '红海': 0.2759713023909607,
'有点': 0.19442262680122022, '节奏': 0.28504415161934643, '战争片':
0.305141973888238, '战争': 0.38941165361568963, '爆破': 0.17280424905747072,
'演员': 0.18291268026465418, '全程': 0.1812416074586381, '湄公河':
0.4937422787186316, '还有': 0.17809860837238478, '个人': 0.2610284731772877,
'黑鹰坠落': 0.21305500057787405, '剧情': 0.21982545428026873, '战狼':
0.611947562855697, '从头': 0.21612064060347713, '文戏': 0.38037163533740115,
'军事': 0.39830147699101087, '好看': 0.1843800611024451, '觉得':
0.18438026420714343, '坦克': 0.2103294696884718, '海军': 0.2780877491487478,
'黄景': 0.2762848729539791, '喜欢': 0.186512622392206, '好莱坞':
0.3041242625437134, '狙击手': 0.4529383170599891,…}
```

从结果可以看到进行词频统计了，但数据中还有“太”“的”等虚词（停用词），这些词在任何场景中都是高频词，并且没有实际的含义，所以要把它们清除。

本程序把停用词放在一个名为 stopwords.txt 的文件中，将数据与停用词进行比对即可。删除停用词的代码如下：

```
keywords={x:keywords[x] for x in keywords if x not in stopwords}
print("删除停用词后",keywords)
```

继续使用 print()语句查看结果，可见停用词已经被清除了。

由于前面只是爬取了第 1 页的评论，所以数据有点少，在最后给出的完整代码中爬取了 10 页的评论，所得数据比较有参考价值。

❸ 用词云进行显示

```
import matplotlib.pyplot as plt
import matplotlib
matplotlib.rcParams['figure.figsize']=(10.0, 5.0)
from wordcloud import WordCloud          #词云包
#指定字体类型、字体大小和字体颜色
wordcloud=WordCloud(font_path="simhei.ttf",background_color="white",
max font size=80,stopwords=stopwords)
word frequence=keywords
myword=wordcloud.fit words(word frequence)
plt.imshow(myword)                       #展示词云图
plt.axis("off")
plt.show()
```

其中，simhei.ttf 用来指定字体，用户可以在百度上输入 simhei.ttf 进行下载，然后放入程序的根目录中。

完整的程序代码如下：

```
import warnings
warnings.filterwarnings("ignore")
import jieba                             #分词包
import jieba.analyse
import numpy                             #numpy 计算包
import re
import matplotlib.pyplot as plt
from urllib import request
from bs4 import BeautifulSoup as bs
import matplotlib
matplotlib.rcParams['figure.figsize']=(10.0, 5.0)
from wordcloud import WordCloud          #词云包
#分析网页函数
def getNowPlayingMovie list():
   resp=request.urlopen('https://movie.douban.com/nowplaying/
   zhengzhou/')
   html data=resp.read().decode('utf-8')
   soup=bs(html data, 'html.parser')
   nowplaying movie=soup.find all('div', id='nowplaying')
   nowplaying movie list=nowplaying movie[0].find all('li', class =
   'list-item')
   nowplaying list=[]
   for item in nowplaying_movie_list:
```

```
            nowplaying dict={}
            nowplaying dict['id']=item['data-subject']
            for tag img item in item.find all('img'):
                nowplaying dict['name']=tag img item['alt']
                nowplaying_list.append(nowplaying_dict)
        return nowplaying list
    #爬取评论函数
    def getCommentsById(movieId, pageNum):  #参数为电影 id 号和要爬取评论的页码
        eachCommentList=[];
        if pageNum>0:
             start=(pageNum-1)*20
        else:
            return False
        requrl='https://movie.douban.com/subject/' + movieId + '/comments'
        +'?' +'start=' + str(start) + '&limit=20'
        print(requrl)
        resp=request.urlopen(requrl)
        html data=resp.read().decode('utf-8')
        soup=bs(html data,'html.parser')
        comment div lits=soup.find all('div', class ='comment')
        for item in comment div lits:
            if item.find all('p')[0].string is not None:
                eachCommentList.append(item.find_all('p')[0].string)
        return eachCommentList
    def main():
        #循环获取第 2 个电影的前 10 页评论
        commentList=[]
        NowPlayingMovie list=getNowPlayingMovie list()
        for i in range(10):                  #前 10 页
            num=i + 1
            commentList temp=getCommentsById(NowPlayingMovie list[1]['id'],
            num)#指定哪部电影。因为索引号从 0 开始，所以是第 2 个电影。numb 是爬取哪一页评论
            commentList.append(commentList_temp)
        #将列表中的数据转换为字符串
        comments=''
        for k in range(len(commentList)):
            comments=comments+(str(commentList[k])).strip()
        #使用正则表达式去掉标点符号
        pattern=re.compile(r'[\u4e00-\u9fa5]+')
        filterdata=re.findall(pattern, comments)
        cleaned comments=''.join(filterdata)
        #使用 jieba 分词进行中文分词
       result=jieba.analyse.textrank(cleaned comments,topK=50,
       withWeight=True)
        keywords=dict()
```

```
    for i in result:
        keywords[i[0]]=i[1]
    print("删除停用词前",keywords)        #{'演员':0.18290354231824632,'大片':
                                         #0.2876433001472282}
    #停用词集合
    stopwords=set(STOPWORDS)
    f=open('./StopWords.txt',encoding="utf8")
    while True:
        word=f.readline()
        if word=="":
            break
        stopwords.add(word[:-1])
    print(stopwords)
    keywords={x:keywords[x] for x in keywords if x  not in stopwords}
    print("删除停用词后",keywords)
    #用词云进行显示
    wordcloud=WordCloud(font path="simhei.ttf",background color="white",
max font size=80,stopwords=stopwords)
    word frequence=keywords
    myword=wordcloud.fit words(word frequence)
    plt.imshow(myword)                    #展示词云图
    plt.axis("off")
    plt.show()
#主函数
main()
```

程序运行后显示的图像如图 20-9 所示。

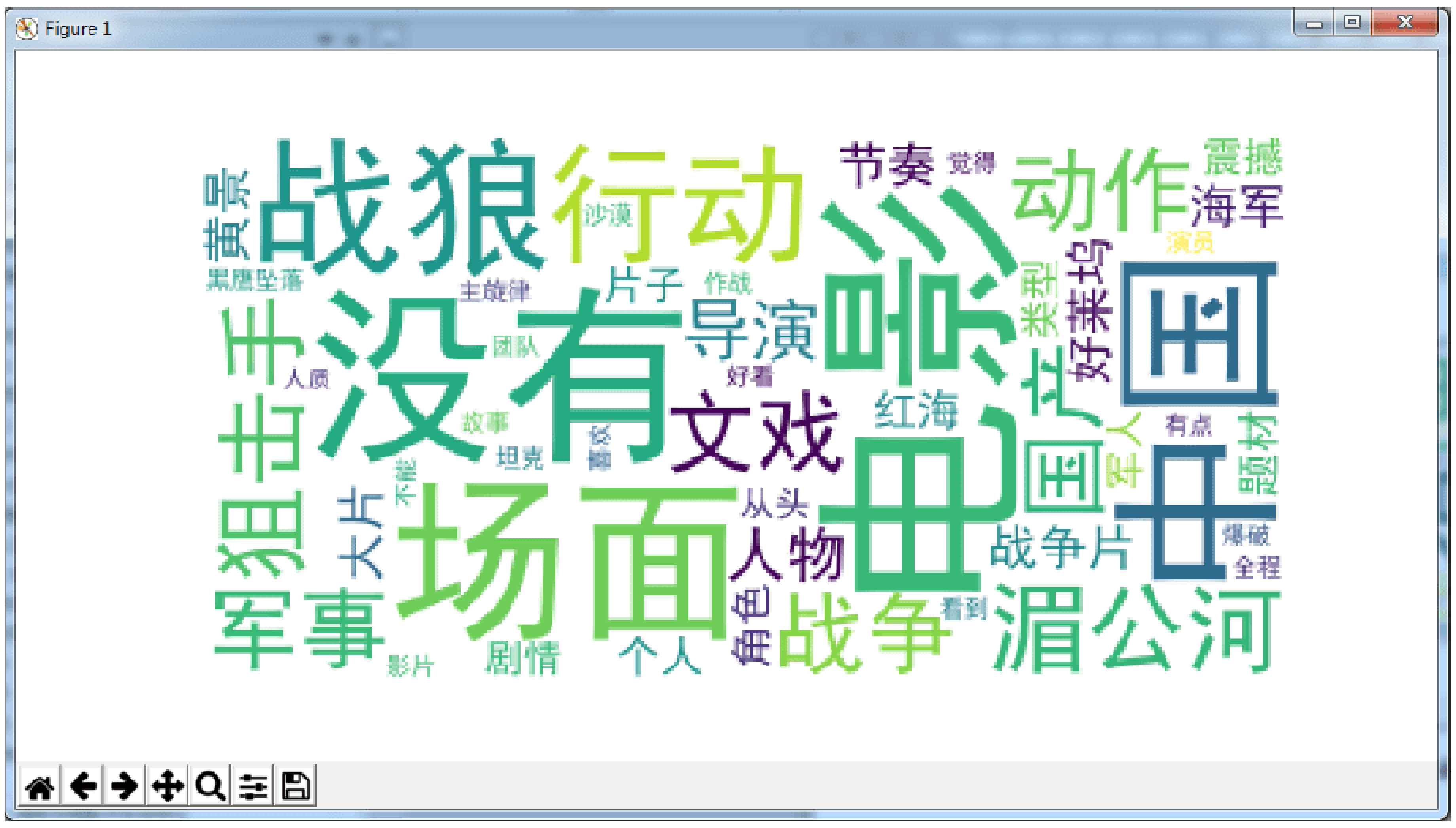


图 20-9　词云显示结果

参 考 文 献

[1] 刘浪. Python 基础教程[M]. 北京：人民邮电出版社，2015.

[2] 江红，余青松. Python 程序设计[M]. 北京：北京交通大学出版社，2014.

[3] 菜鸟教程. Python3 教程. http://www. runoob. com/python3.

[4] 廖雪峰. Python 教程. http://www. liaoxuefeng. com/.

[5] 陈锐，李欣，夏敏捷. Visual C#经典游戏编程开发[M]. 北京：科学出版社，2011.

[6] 郑秋生，夏敏捷. Java 游戏编程开发教程[M]. 北京：清华大学出版社，2016.

[7] 夏敏捷. 校园网 Web 搜索引擎的设计与实现[J]. 中原工学院学报，2011.

图书资源支持

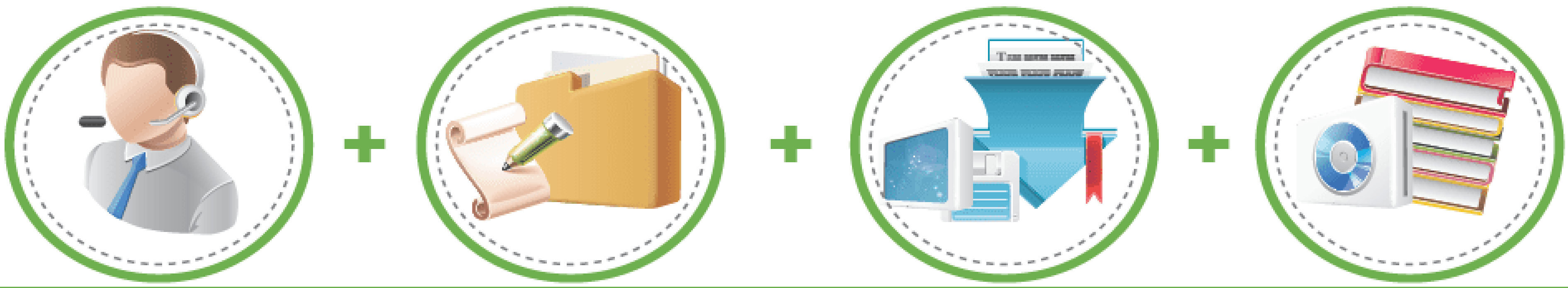
感谢您一直以来对清华版图书的支持和爱护。为了配合本书的使用，本书提供配套的资源，有需求的读者请扫描下方的“书圈”微信公众号二维码，在图书专区下载，也可以拨打电话或发送电子邮件咨询。

如果您在使用本书的过程中遇到了什么问题，或者有相关图书出版计划，也请您发邮件告诉我们，以便我们更好地为您服务。

全国高等院校应用型创新规划教材 计算机系列

Python 数据分析基础

余本国 编著



- 由浅入深，配有大量示例代码，有利于尽快进入角色
- 理论与实际应用紧密结合，通过案例讲解，突出实用性
- 从实例出发，结合课后练习，让读者少走弯路
- 配备免费教学资源——电子课件、习题答案

清华大学出版社

全国高等院校应用型创新规划教材•计算机系列

Python 数据分析基础

余本国　编　著

清华大学出版社
北　京

内 容 简 介

Python 是由 Guido van Rossum 于 20 世纪 80 年代末和 90 年代初，在荷兰国家数学和计算机科学研究所设计出来的。它是一种面向对象的、用途非常广泛的编程语言，具有非常清晰的语法特点，适用于多种操作系统。目前 Python 在国际上非常流行，正在得到越来越多的应用。

Python 可以完成许多任务，功能非常强大，其利用 Pandas 处理大数据的过程，由于 Pandas 库的使用能够很好地展现数据结构，成为近来 Python 项目中经常使用的热门技术，并且 R 和 Spark 对 Python 都有很好的调用接口，甚至在内存使用方面都有优化。

本书根据作者多年教学经验编写，条理清楚，内容深浅适中，尽量让读者从实例出发，结合课后练习，少走弯路。本书涉及的内容主要包括 Python 数据类型与运算、流程控制及函数与类、Pandas 库的数据处理与分析等。在本书的最后，还附带了一些文件读写、网络爬虫、矩阵计算等最基本的内容。

本书可以作为本科生、研究生以及科研人员学习 Python 的基础教材。

图书在版编目(CIP)数据

Python 数据分析基础/余本国编著. —北京：清华大学出版社，2017
(全国高等院校应用型创新规划教材・计算机系列)
ISBN 978-7-302-47890-4

Ⅰ. ①P…　Ⅱ. ①余…　Ⅲ. ①软件工具—程序设计—高等学校—教材　Ⅳ. ①TP311.561

中国版本图书馆 CIP 数据核字(2017)第 184452 号

责任编辑：秦　甲
封面设计：杨玉兰
责任校对：宋延清
责任印制：沈　露
出版发行：清华大学出版社
网　　址：http://www.tup.com.cn, http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编：**100084
社 总 机：010-62770175　　**邮　　购：**010-62786544
投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn, 010-62791865
印 装 者：三河市金元印装有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张：**14.5　　**字　数：**279 千字
版　　次：2017 年 8 月第 1 版　　**印　次：**2017 年 8 月第 1 次印刷
印　　数：1～2500
定　　价：39.00 元

产品编号：071752-01

前　　言

在写作本书的时候，国内大多数参考书还是 Python 2.7 版本，为了给在校大学生开设这门 Python 课程，我们选择了 Python 3.x，毕竟 Python 3.x 才是未来。与其让学生们从 Python 2.7 开始学，还不如直接从 Python 3.x 上手，以掌握更加完善的知识。

作者通过近三轮的教学，对 Python 3.x 的基础知识进行了筛选和总结，特编写此书，希望能够给准备使用 Python 的读者提供一些方便。

本书由浅入深，比较适合那些从未接触过计算机语言的读者。每章配有大量的示例代码，希望读者在使用本书的时候，能够尽可能自己敲代码，少用复制粘贴的方法，这样有利于读者尽快进入“角色”，毕竟“拷贝得来终觉浅”。

本书的前 3 章是 Python 的基础知识；第 4 章是利用 Pandas 库对数据进行处理、分析以及实现数据可视化；在第 5 章还列出了 Python 对文件的读取、存储方法，对网络爬虫、矩阵运算也做了简单的介绍。

作者在编写本书的过程中，得到了 Python 工程师齐伟的帮助。在开设这门课的时候，齐伟通过视频的形式与我们一起分享了 Python 开发经验。本书在完稿时，得到了研究生闫青、陈文华、马秀、樊宇凯和卢超在文字校对上的帮助。

最后感谢广大读者选择了本书。作者 E-mail：yubg@nuc.edu.cn，QQ：120487362。欢迎各位读者批评指正。

编　者

目录

第 1 章

Python 简介

Python 语言是一种简单而又功能强大的编程语言。通过学习，你会发现，Python 语言注重的是如何解决问题，而不是编程语言的语法和结构。

Python 的官方介绍是：Python 是一种简单易学、功能强大的编程语言，它有高效率的高层数据结构，能简单而有效地实现面向对象编程。

Python 简洁的语法和对动态输入的支持，再加上解释性语言的本质，使得它在大多数平台上的许多领域中都是一个理想的脚本语言，特别适用于快速的应用程序开发。不需要任何编程基础，完全可以从零开始学习。

注意： Python 语言的创造者 Guido van Rossum 是根据英国广播公司的节目“蟒蛇飞行马戏”来命名这个语言的，Python 的英文本意是“巨蛇，大蟒”。

Python 确实是一种十分精彩且强大的语言。它合理地结合了高性能及使得编写程序简单有趣的特色。

Python 的缺点：前后版本不兼容。这确实是让新、老学习人员感到有点头痛的事情。

因为前后版本不兼容，导致许多人为选择 Python 2.x 还是 Python 3.x 发愁。本书推荐使用 Python 3.x。

的确，当前还有相当多的程序员在使用 Python 2.7，不过 Python 3.x 才是 Python 发展的未来，这就像 Windows 7 和 Windows 10 谁是未来一样。

我们发现，Python 3.x 中的新特性确实很妙，很值得进行深入学习。

其实，我们也不用担心，如果了解了 Python 3.x，则 Python 2.7 的代码阅读起来是根本不成问题的。

1.1 安装 Python

Windows 用户可以访问 https://python.org/download，从网站中下载最新的版本，大小约为 27.4MB。与其他大多数语言相比，Python 的安装包算是十分紧凑的，其安装过程与其他 Windows 软件类似。

在本书即将完成的时候，我们使用的是最新版 Python 3.5.1，所使用的计算机系统为 Windows 10。

安装 Python 很简单，双击 python-3.5.1.exe，勾选 Add Python 3.5 to PATH，再单击 Install Now 即可，如图 1-1 所示，其下方已经显示了安装路径。安装完毕后，会显示安装成功界面，最后单击 Close 按钮就可以使用了。

安装完成后，在“开始”菜单中会显示安装目录，如图 1-2 所示。当我们要编写代码时，直接选择 IDLE 命令即可。

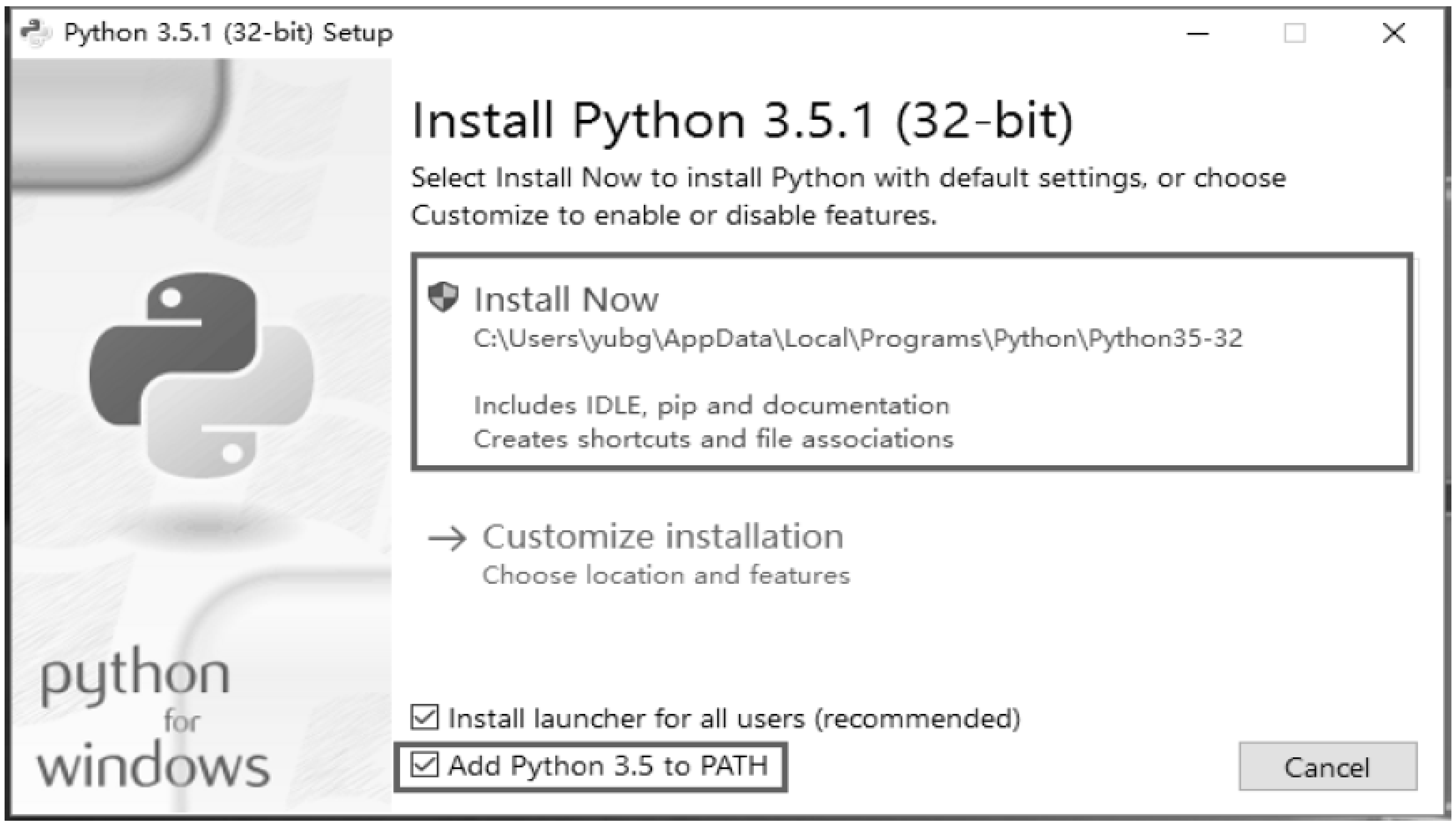


图 1-1　安装界面



图 1-2　“开始”菜单中的安装目录

如果是在 Windows 7 系统中，安装完毕后，还要进行环境的配置(以下是在 Windows 7 系统上安装的 Python 3.3 版本，3.5 版本的安装方法大致相同)，具体方法如下。

打开“控制面板”窗口，单击“系统”图标，打开“系统”窗口，在左侧单击“高级系统设置”图标，将弹出“系统属性”对话框。在该对话框中单击“环境变量”按钮，将弹出“环境变量”对话框，在“系统变量”列表框中选择 Path 选项，然后单击“编辑”按钮，在弹出的“编辑系统变量”对话框中编辑 Path 变量。把“**;C:\Python33**”添加到变量值的末尾，如图 1-3 所示。当然，前提是 Python 已经正确地安装在 C 盘的根目录下，即 C 盘中已经存在 Python33 文件夹。

然后在 DOS Shell 命令提示符下输入“python”，如果看到如图 1-4 所示的信息，就说明 Python 已经安装成功了。

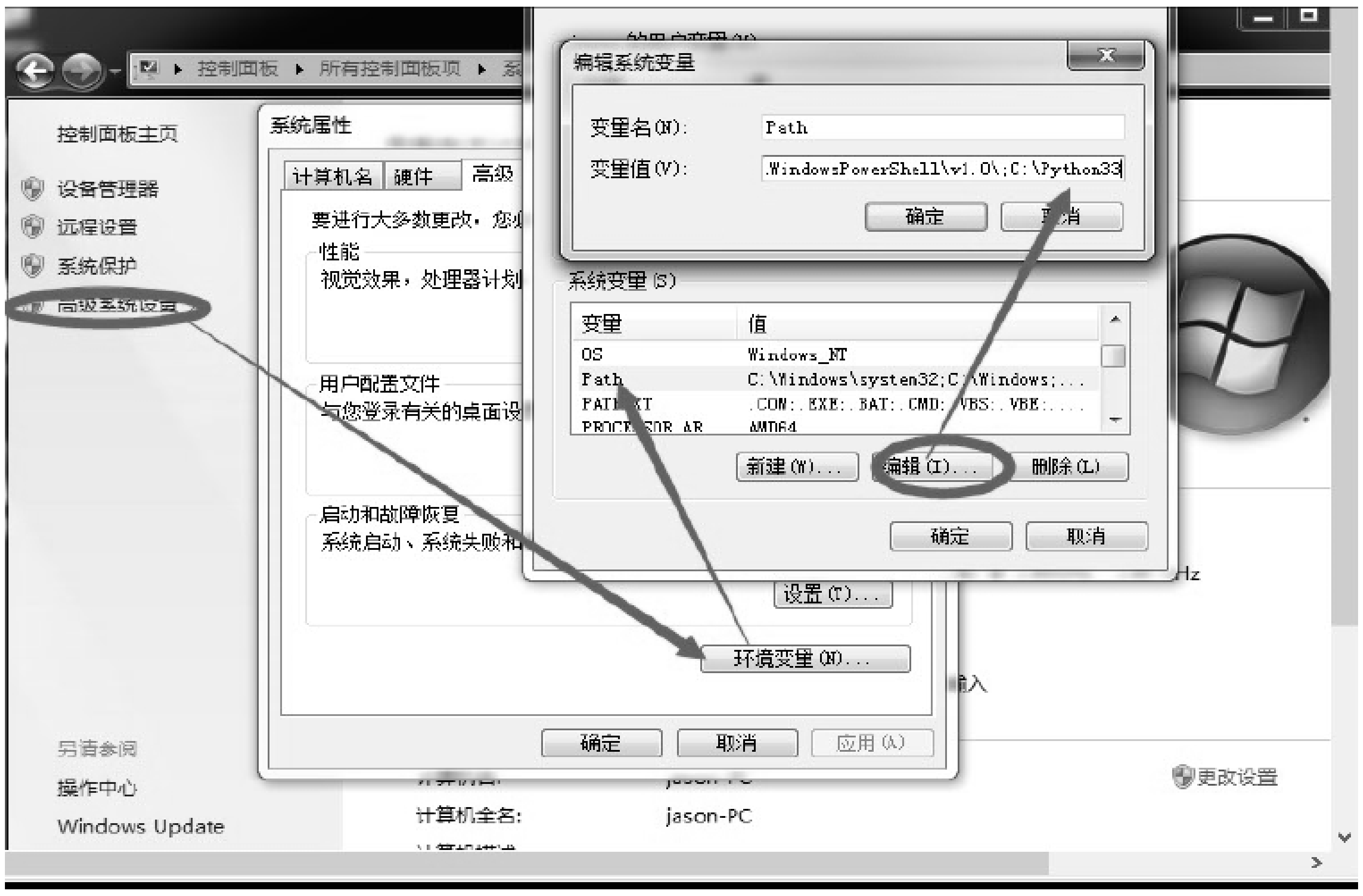


图 1-3　系统变量的设置

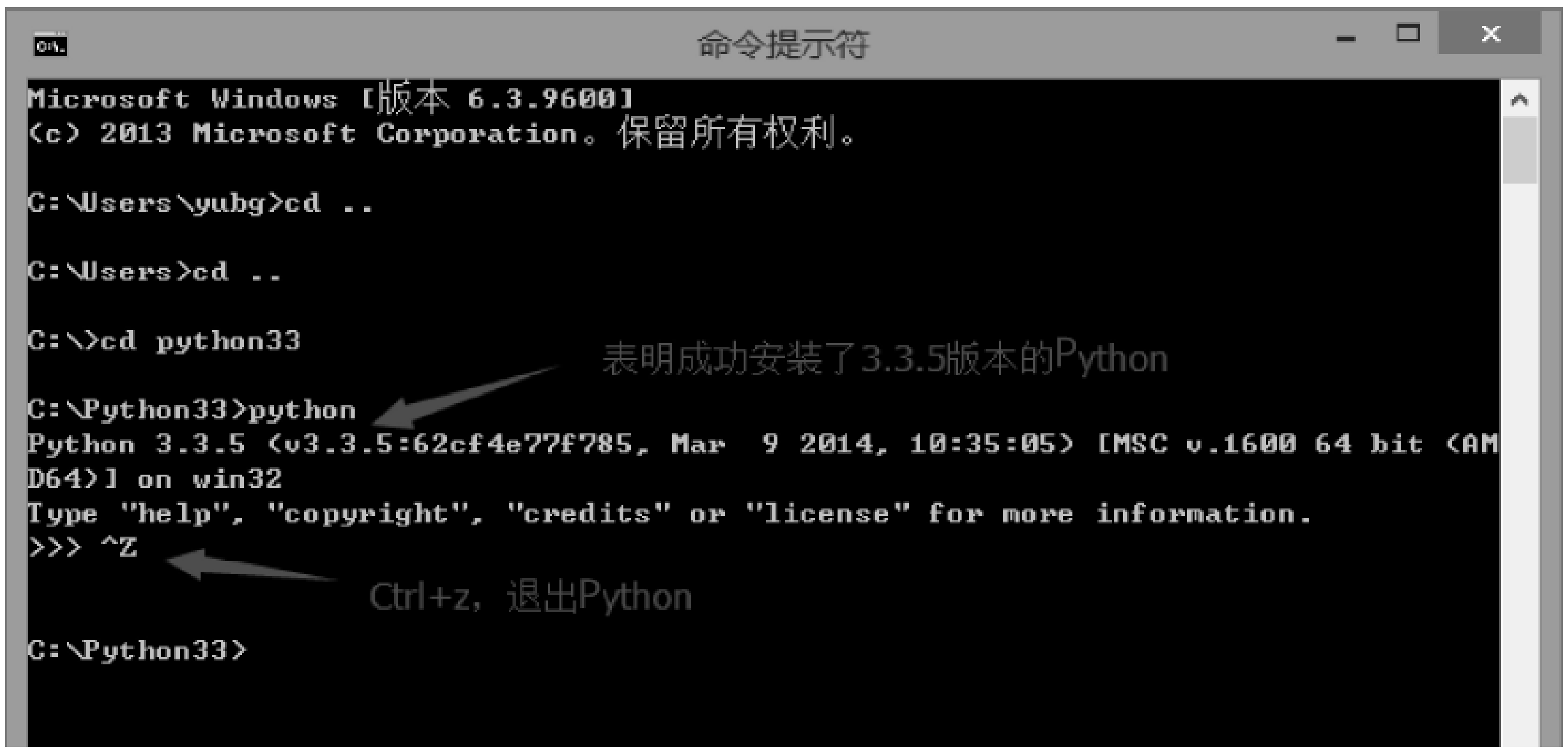


图 1-4　在命令提示符下测试 Python 安装

图 1-4 中显示的是在 C 盘下已安装 Python，目录为 C:\Python33。如果与此不一致，需要先下载并安装 Python。

后面还会介绍 Python 的其他安装方法，目的是为了避免安装复杂的 Python 数据包，如 pandas、numpy 等。

关于 Python 下载和学习的网站很多，例如：

- http://freelycode.com/fcode/downloadinstall?listall=True
- www.freelycode.com
- http://pythontutor.com/

综上所述，对于 Windows 系统，要安装 Python，只须下载安装程序，然后双击它就可以了，是非常简单的。从现在起，我们假设已经在计算机系统里安装了 Python 3.5。

打开 Python 的 IDLE，启动 Python 解释器。

我们在>>>提示符后面输入 print('Hello World')，然后按 Enter 键，应该可以看到输出了 Hello World，如图 1-5 所示。

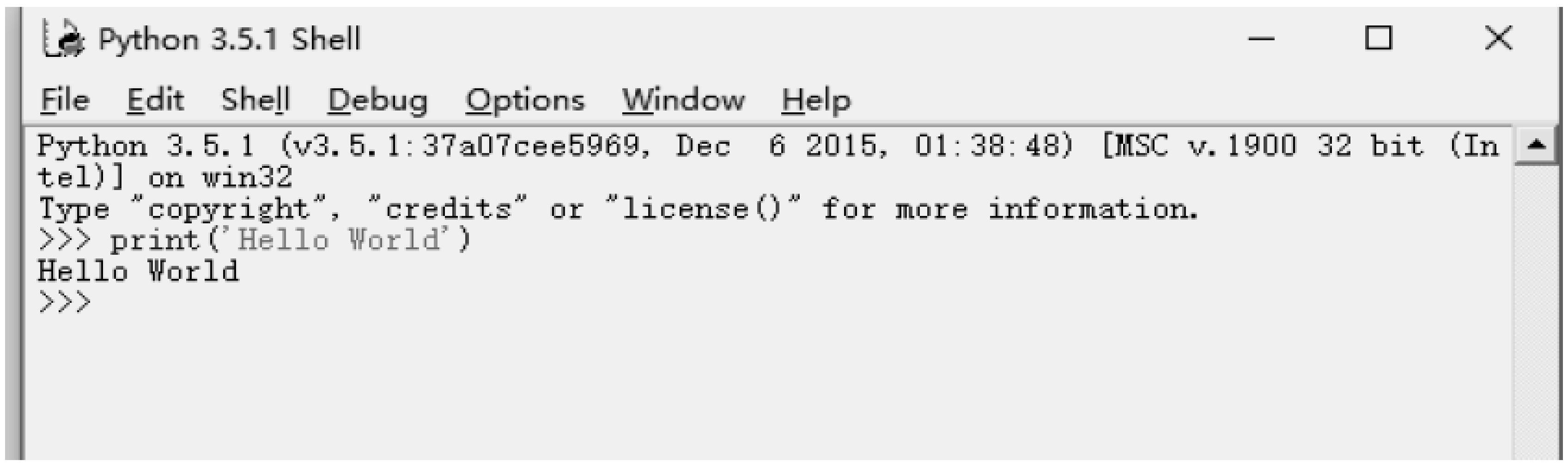


图 1-5　IDLE 界面

注意： 此处的>>>为系统自动显示的提示符，不需要人为地输入，而程序中所涉及的括号()、引号''等，都需要在英文半角状态下输入。

1.2　Python 2 和 Python 3 的区别

本节我们讲解 Python 2 和 Python 3 的主要区别。

1．性能

Python 3.0 运行 PyStone Benchmark 的速度比 Python 2.5 慢 30%。Guido 认为 Python 3.0 有极大的优化空间，在字符串和整型操作上可以取得很好的优化结果。目前 Python 3.5 版本的性能已经优于版本 2.7。

2．编码

Python 3.x 源码文件默认使用 utf-8 编码，这就使得以下代码是合法的，但在 Python 2.x 中是不可思议的事情，对于中国人来说是个“福音”：

```
>>> 中国 = 'china'
>>> print(中国)
china
>>>
```

3．语法

(1)　去除了不等号<>，全部改用!=。

(2)　关键词加入 as 和 with，还有 True、False、None。

(3)　整型除法返回浮点数，要得到整型结果，须使用//。

(4) 去除 print 语句，加入 print()函数实现相同的功能。同样地，还有 exec 语句，已经改为 exec()函数。

例如：

```
print "The answer is", 2*2          # 2.x
print("The answer is", 2*2)         # 3.x
```

(5) 改变了顺序操作符的行为，例如 x<y，当 x 和 y 类型不匹配时抛出 TypeError，而不是返回随机的 bool 值。

(6) 输入函数有改变，删除了 raw_input，用 input 代替。

读取键盘输入的方法如下：

```
guess = int(raw_input('Enter an integer : '))          # 2.x
guess = int(input('Enter an integer : '))              # 3.x
```

(7) 删除了 cmp()比较函数。

4．数据类型

(1) Python 3.x 去除了 long 类型，现在只有一种整型——int，但它的行为就像 2.x 版本的 long。

(2) dict 的.keys()、.items 和.values()方法返回迭代器，而先前的 iterkeys()等函数都被废弃。同时去掉的还有 dict.has_key()，用 in 替代。

5．异常

(1) 所有异常都从 BaseException 继承，并删除了 StandardError。

(2) 去除了异常类的序列行为和.message 属性。

(3) 用 raise Exception(args)代替 raise Exception, args 语法。

(4) 捕获异常的语法改变，引入了 as 关键字来标识异常实例。在 Python 2.5 中：

```
>>> try:
   raise NotImplementedError('Error')
except NotImplementedError, error:
   print error.message

Error
>>>
```

在 Python 3.0 中：

```
>>> try:
raise NotImplementedError('Error')
except NotImplementedError as error:     #注意这里的 as
```

```
print(str(error))
Error
>>>
```

6．模块变动

(1)　移除了 cPickle 模块，使用 pickle 模块代替。

(2)　移除了 imageop 模块。

(3)　移除了 audiodev、bastion、bsddb185、exceptions、linuxaudiodev、md5、mimify、MimeWriter、popen2、rexec、sets、sha、stringold、strop、sunaudiodev、timing 和 xmllib 模块。

(4)　移除了 bsddb 模块(单独发布，可以从 http://www.jcea.es/programacion/pybsddb.htm 获取)。

(5)　移除了 new 模块。

(6)　os.tmpnam()和 os.tmpfile()函数被移动到 tmpfile 模块下。

7．其他

(1)　xrange()改名为 range()，要想使用 range()获得一个 list，必须显式调用：

```
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>>
```

(2)　bytes 对象不能 hash，也不支持 b.lower()、b.strip()和 b.split()方法，但对于后两者，可以使用 b.strip(b'\n\t\r \f')和 b.split(b' ')来达到相同的目的。

(3)　zip()、map()和 filter()都返回迭代器。

(4)　file 类被废弃，在 Python 2.5 中：

```
>>> file
<type 'file'>
>>>
```

在 Python 3.x 中：

```
>>> file
Traceback (most recent call last):
File "<pyshell120>", line 1, in <module>
file
NameError: name 'file' is not defined
>>>
```

本 章 小 结

本章主要学习了 Python 的安装和 IDLE 的启用，以及了解了 Python 2.7 和 Python 3.x 之间的差异。

练　　习

(1) 请将 IDLE 的 Shell 界面字体调试成 18 号等线 light 字体。

(2) 在 Shell 编辑器的>>>后输入 help()，查看本机安装的 Python 版本信息。

第 2 章

Python 数据类型与运算

我们先了解 Python 的几个语法常识。

1．代码注释方法

(1) 在一行中，“#”后的语句不再执行，而表示被注释。

(2) 如果要进行大段的注释，可以使用三个单引号(’’’)或者双引号(”””)将注释内容包围。单引号和双引号在使用上没有本质的差别。

【例 2-1】三个双引号注释段落：

```
# -*- coding: utf-8 -*-
"""
Created on Sun Mar 13 21:20:06 2016
@author: yubg
"""
lis=[1,2,3]
for i in lis:   #半角状态冒号不能少，下一行注意缩进
    i+=1
print(i)
```

本例不需要上机操作，仅为展示用法。

2．用缩进来表示分层

Python 不像 C 语言那样用{}来表示语句块，而是通过让代码缩进 4 个空格来表示分层，当然也可以使用 Tab 键，但不要混合使用 Tab 键和空格来进行缩进，否则会使程序在跨平台时不能正常工作，官方推荐的做法是使用四个空格。

一般来说，行尾遇到“:”就表示下一行缩进的开始，如例 2-1 中的“for i in lis”行尾有冒号，下一行的“i+=1”就需要缩进四个空格。

3．语句断行

一般来说，Python 中的一条语句占一行，在每条语句的结尾处不需要使用分号(;)。但在 Python 中也可以使用分号，表示将两条简单语句写在一行。但如果一条语句较长，要分几行来写，可以使用“\”来进行换行。分号还有个作用，使用在一行语句的末尾，表示对本行语句的结果不打印输出。一般地，系统能够自动识别换行，如在一对括号中间或三引号之间均可换行。例如下面代码中的第三行较长，若要对其分行，则必须在括号内进行(包括圆括号、方括号和花括号)：

```
from pandas import DataFrame   #导入模块中的函数，后面再讲
from pandas import Series
df = DataFrame({'age':Series([26,85,64]),'name':Series(['Ben','Joh','Jef'])})
print(df)
```

分行后的第二行一般空四个空格，在 3.5 版本中已经优化，可以不空四个空格，但是在较低的 3.x 版本中不空四个空格会报错。

```
from pandas import DataFrame
from pandas import Series
df = DataFrame({'age':Series([26,85,64]),   #此语句分成了两行
    'name':Series(['Ben','Joh','Jef'])})
print(df)
```

4. print()的作用

print()会在输出窗口中显示一些文本或结果，便于验证和显示数据。

5. 使用转义符

如果需要在一个字符串中嵌入一个引号，该如何操作？

有两种方法：可以在引号前加反斜杠(\)，或者用不同的引号包围这个引号。

例如：

```
>>>s1='I\'am a boy. '  #可以使用转义符\
>>>print(s1)
I'am a boy.

>>>s2="I'am a boy. "   #也可以用不同的引号包围起来，此处用双引号是为了区分单引号
>>>print(s2)
I'am a boy.
>>>
```

转义符详见本章 2.2.5 小节的内容。

2.1 数据类型

Python 总共有 6 种数据类型，分别是数字型(Numbers)、字符串型(String)、列表型(List)、元组型(Tuple)、集合型(Sets)和字典型(Dictionaries)。

数字型又可划分为整数型(int)、浮点型(float)、布尔型(bool)和复数型(complex)。

在 Python 中有 4 种类型的数——整数、长整数、浮点数和复数。

例如，2 是一个整数的例子。

长整数不过是大一些的整数。

3.23 和 52.3E-4 是浮点数的例子，E 标记表示 10 的幂。52.3E-4 表示 52.3×10^{-4}。

(-5+4j)和(2.3-4.6j)表示的是复数。

查看数据类型的方法是：

```
>>> type(变量名)
```

例如：

```
>>> a = 1  #int
>>> type(a)
<class 'int'>
>>> b = True  #boolean
>>> type(b)
<class 'bool'>
>>> c = 4+3j  #complex
>>> type(c)
<class 'complex'>
>>> d = 3.14   #float
>>> type(d)
<class 'float'>
>>>
```

2.2 运算符与功能命令

2.2.1 算数运算符

算数运算符如表 2-1 所示。

表 2-1 算术运算符

操 作 符	描　述	例子(a=10,b=20)
+	加法，两个对象相加	a + b = 30
-	减法，一个数减去另一个数	a - b = -10
*	乘法，两数相乘或返回一个被重复若干次的字符串	a * b = 200
/	除法，b 除以 a	b / a = 2
%	取模，返回除法的余数	b % a = 0
**	指数，返回 a 的 b 次幂	a**b=100000000000000000000，10 的 20 次幂
//	取整除，返回商的整数部分	9//2 = 4，而 9.0//2.0 = 4.0

2.2.2 比较运算符

比较运算符如表 2-2 所示。

表 2-2　比较运算符

运 算 符	描　述	例子(a=10,b=20)
==	检查两个操作数的值是否相等，若相等则条件变为真	(a == b)不为 true
!=	检查两个操作数的值是否相等，若不相等则条件变为真	(a != b)为 true
>	检查左操作数的值是否大于右操作数的值，若大于则条件为真	(a > b)不为 true
<	检查左操作数的值是否小于右操作数的值，若小于则条件为真	(a < b)为 true
>=	检查左操作数的值是否大于或等于右操作数的值，若大于等于则条件为真	(a >= b)不为 true
<=	检查左操作数的值是否小于或等于右操作数的值，若小于等于则条件为真	(a <= b)为 true

2.2.3　赋值运算符

赋值运算符如表 2-3 所示。

表 2-3　赋值运算符

运算符	描　述	示　例
=	简单的赋值运算符，赋值从右侧操作数到左侧操作数	c = a + b 相当于将 a + b 赋值给 c
+=	加法赋值运算符，左操作数和右操作数和的结果赋给左操作数	c += a 相当于 c = c + a
-=	减赋值运算符，左操作数减去右操作数，并将结果赋给左操作数	c -= a 相当于 c = c - a
*=	乘法赋值运算符，左操作数与右操作数的乘积赋给左操作数	c *= a 相当于 c = c * a
/=	除法赋值运算符，左操作数除以右操作数的结果赋给左操作数	c /= a 相当于 c = c / a
%=	模赋值运算符，左操作数与右操作数模的结果赋给左操作数	c %= a 相当于 c = c % a
**=	指赋值运算符，左操作数的右操作数指数之值赋给左操作数	c **= a 相当于 c = c ** a
//=	地板除，左操作数地板除右操作数，将结果赋给左操作数	c //= a 相当于 c = c // a

【例 2-2】各类运算示例：

```
>>> print(1+9)                    # 加法
10
>>> print(1.3-4)                  # 减法
```

```
-2.7
>>> print(3*5)                    # 乘法
15
>>> print(4.5/1.5)                # 除法
3.0
>>> print(3**2)                   # 乘方
9
>>> print(10%3)                   # 求余数
1
>>> print(5==6)                   # 相等
False
>>> print(8.0!=8.0)               # 不相等
False
>>> print(3<3, 3<=3)              # <小于; <=小于等于
False True
>>> print(4>5, 4>=0)              # > 大于; >= 大于等于
False True
>>> print(5 in [1,3,5])           # 5是list [1,3,5]的一个元素
True
>>> print(True and True, True and False)  # and，两者都为真才是真
True False
>>> print(True or False)          # or，“或”运算，其中之一为真即为真
True
>>> print(not True)               # not，“非”运算，取反
False
>>> divmod(5,2)                   # 表示5除以2，返回了商和余数
(2, 1)
>>> 5/2
2.5
>>>a = 1.5
>>>b = 1
>>>a += b
>>>print(a)
2.5
>>>
```

说明：在 Python 3.5 之前的版本中，当两个整数相除时，其结果取商的整数部分(不是四舍五入)；当除数和被除数之一或两者都是浮点数时，其结果才是浮点数。如计算 5/2 时，得到的结果不是 2.5，而是 2。但在 Python 3.5 版本中已经做了优化，其结果显示为 2.5。若是 3.5 之前的版本，也可以用除法模块__future__中的 division，如下所示：

```
>>> from __future__ import division
>>> 5/2
```

```
2.5

>>> round(1.234567,2)
1.23
>>>
```

但是也有例外：

```
>>> round(2.235,2)
2.23
>>>
```

结果为什么不是 2.24？因为这是由浮点数引起的精确度问题，后续章节将详细介绍 decimal 模块对此类问题的处理方法(from decimal import Decimal)：

```
>>> from decimal import Decimal
>>> from decimal import localcontext
>>> a = Decimal('1.3')
>>> b = Decimal('1.7')
>>> print(a / b)
0.7647058823529411764705882353

>>> with localcontext() as ctx:
ctx.prec = 3
print(a / b)

0.765
>>>
```

思考：为什么>>>print('I love i-nuc.com ' * 5)可以正常执行，而>>>print('I love i-nuc.com ' + 5)却报错？

运算类型不一致。在 Python 中不能把两个完全不同的东西加在一起，比如说数字和文本。正是这个原因，print('I love i-nuc.com ' + 5)才会报错。这就像在说“我的体重 67 公斤加上我的身高 170cm 是多少”一样没有多大意义！不过，文本乘以一个整数来翻倍就具有一定的意义了，print('I love i-nuc.com ' * 5)意思就是将“I love i-nuc.com ”这个字符串接连输出 5 次。

2.2.4　常量与变量

常量有数值、字符、逻辑真假，如 12、yy、“12”、true、false。

变量是可以给它赋值的量，如 a=3。

变量只能由数字、字母和下划线构成，但不能以数字开头，而且变量区分大小写；以

下划线开头的变量有特殊含义；变量名不能含有空格和标点符号，如括号、引号、逗号、斜线、反斜线、冒号、句号和问号；在 Python 3.x 中，变量也可以是中文。例如：

```
>>> 中国 = "I love you"
>>> list_1 = [1,2,3]
>>> List = [5,6,7]
>>> list_1
[1, 2, 3]
>>> List
[5, 6, 7]
>>>
```

2.2.5 字符串

字符串是字符的序列。字符串基本上就是一组单词。字符串需用单引号(‘’)或者双引号(“”)括起来。

(1) **使用单引号(’)**：可以用单引号指示字符串，就如同‘Quote me on this’这样。所有的空白，即空格和制表符都照原样保留。

(2) **使用双引号(”)**：双引号与单引号中的字符串在使用上完全相同，例如“What’s your name?”。

(3) **使用三引号(’’’或”””)**：利用三引号，可以指示一个多行的字符串。可以在三引号中自由地使用单引号和双引号。

(4) **转义符**：假设想要在一个字符串中包含一个单引号(’)，例如字符串 What’s your name?，就不能用‘What’s your name?’来表示，因为这里有三个单引号，Python 不知从何处开始，何处结束。这时，可以通过转义符来完成这个任务，用\’来表示单引号——注意这里的反斜杠。可以把字符串表示为‘What\’s your name?’。又如：

```
>>> s = 'Yes,he doesn\'t'
>>> print(s, type(s), len(s))       #len()函数用于查看 s 的长度
Yes,he doesn't <class 'str'> 14
>>>
```

还可以用另一个表示方法，即用双引号，以便与字符串内的单引号区分开，即“What’s your name?”。

类似地，要在双引号字符串中使用双引号时，可以借助于转义符。另外，也可以用转义符“\\”来指示反斜杠“\”。

值得注意的是，在一个字符串中，行末单独一个反斜杠表示字符串在下一行继续，而不是开始一个新的行。例如：

```
"This is the first sentence.\
This is the second sentence."
```

等价于：

```
"This is the first sentence. This is the second sentence."
```

转义字符的种类如表 2-4 所示。

表 2-4　转义字符及其描述

转义字符	描　述
\(在行尾时)	续行符
\\	反斜杠符号
\’	单引号
\”	双引号
\a	响铃
\b	退格(BackSpace)
\e	转义
\000	空
\n	换行
\v	纵向制表符
\t	横向制表符
\r	回车
\f	换页
\oyy	八进制数 yy 代表的字符，例如，\o12 代表换行
\xyy	十进制数 yy 代表的字符，例如，\x0a 代表换行
\other	其他的字符以普通格式输出

(5)　**自然字符串**：如果想要指示某些不需要如转义符那样特别处理的字符串，则需要指定一个自然字符串。自然字符串通过在字符串前添加前缀 r 或 R 来指定。

例如 a=r“Newlines are indicated by \n”，读者可自行测试。

又如：

```
>>> print('C:\some\name')    # \n 表示成了换行符
C:\some
ame
>>> print(r'C:\some\name')
C:\some\name
>>>
```

(6)　**Unicode 字符串**：Unicode 是书写国际文本的标准方法。如果想要用其他语言，如

北印度语或阿拉伯语写文本，那么就需要有一个支持 Unicode 的编辑器。类似地，Python 允许处理 Unicode 文本——只需要在字符串前加上前缀 u 或 U。例如，u“This is a Unicode string.”。

(7) **字符串是不可变的**：这意味着一旦创造了一个字符串，就不能再改变它。虽然这看起来像是一件坏事，但实际上它不是。我们将会在后面的程序中看到为什么说它不是一个缺点。

字符串可以通过“+”运算符串连在一起，或者用“*”运算符重复。例如：

```
>>> print('str'+'ing', 'my'*3)
string mymymy
>>>
```

2.2.6 字符串索引与切片

定义字符串变量并赋值的例子如下：

```
>>> str = "Hello My friend"
```

字符串是一个整体。如果想直接修改字符串，是不可能做到的，但我们能够读出字符串中的某一部分。

1. 字符串的索引

给出一个字符串，可输出其任意一个字符。Python 中的字符串有两种索引方式：第一种是从左往右，从 0 开始依次增加；第二种是从右往左，从-1 开始依次减少。例如：

```
>>> word = 'Python'
>>> print(word[0])
P
>>> print(word[-1], word[-6])
n P
>>>
```

2. 字符串的切片

切片(也叫分片)就是从给定的字符串中分离出部分内容。

(1) Python 中用冒号分隔两个索引，形式为“变量[头下标:尾下标]”，截取的范围是左闭右开，即不包含尾下标的字符，并且两个索引都可以省略。例如：

```
>>> str = "Hello My friend"
>>> print(str[1:4])
ell
```

```
>>> print(str[:-7])
Hello My
>>> print(str[5:])
 My friend
>>> print(str[:])
Hello My friend
>>>
```

(2) 切片的扩展形式为“str[I:J:K]”，从 I 到 J-1，每隔 K 个元素索引一次，如果 K 为负数，就是按从右往左索引。例如：

```
>>> str = "Hello My friend"
>>> print(str[2:7:2])
loM
>>> print(str[2:7:1])
llo M
>>>
```

(3) 字符串包含判断操作符为 in，not in。例如：

```
>>> str = "Hello My friend"
>>> "He" in str
True
>>> "she" not in str
True
>>> str.find('o')          #字符串模块提供的查找方法
4
>>>
```

(4) ord 函数将字符转化为对应的 ASCII 码值，而 chr 函数将数字转化为字符。例如：

```
>>> print(ord('a'))
97
>>> print(chr(97))
a
>>>
```

(5) 处理字符串的内置函数：

```
len(str)                   #串长度
max('abcxyz')              #寻找字符串中最大的字符
min('abcxyz')              #寻找字符串中最小的字符
```

(6) string 的转换：

```
int(str)         #变成整型，如 int("12")的结果为 12，是数值型
```

string 模块还提供了很多方法，例如：

```
S.find(substring, [start [,end]]) #可指范围查找子串，返回索引值，否则返回-1
S.rfind(substring,[start [,end]]) #反向查找
S.index(substring,[start [,end]]) #同 find，只是找不到会产生 ValueError 异常
S.rindex(substring,[start [,end]])#同上反向查找
S.count(substring,[start [,end]]) #返回找到子串的个数
S.capitalize()       #首字母大写
S.lower()           #转小写
S.upper()           #转大写
S.swapcase()         #大小写互换
S.split()    #将 string 转 list，默认以空格切分，也可以指定字符切分
```

2.2.7 输入和输出

1. print

print 是一个常用函数，其功能就是输出括号中的字符串(在 Python 2.x 中，print 格式写成 print ‘Hello World!’)。

print 可以有多个输出，以逗号分隔。例如：

```
>>> a=10
>>> print(a,type(a))
10 <class 'int'>
>>>
```

若要将多个结果打印在一行，并以逗号分隔，可以在 print 中添加 end=‘,’，例如 print(test_list[i], end=‘,’)，后面将会用到。

2. input

input 函数将用户输入的内容作为**字符串**的形式返回，就算输入的是数字，但返回的“数字”的类型也是字符串型。

【例 2-3】input 输入：

```
>>> a = input("Input: ")
Input: 12
>>> a
'12'
>>> type(a)
<class 'str'>
>>> b=input('请您输入:')
请您输入:abc
```

```
>>> b
'abc'
>>> type(b)
<class 'str'>
>>> c = int(input("请您输入数字："))
请您输入数字：231
>>> type(c)
<class 'int'>
>>> d=int(input('请输入字符：'))
请输入字符：acd
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
    d=int(input('请输入字符：'))
ValueError: invalid literal for int() with base 10: 'acd'
>>>
```

如果要想获取数字，可以使用 int 函数将接收进来的“数字”字符转化为数字。例如：

```
x = int(input("输入："))
```

但需要注意，不可以输入字符型，如输入字母则会报错，如例 2-3 最后输入的 acd。

2.2.8　原始字符串

先看个例子：

```
>>> d="c:\news"
>>> d
'c:\news'
>>> print(d)
c:
ews
>>>
```

执行 print(d)语句时，出现的不是 c:\news，而是把其中的\n 看成了换行符。

为了避免在 print 的时候出现上述的歧义现象，可以引入转义符，即 d=“c:\\news”，但若输出的路径较长，为了方便起见，可以在引号的前面加上 r，表示 r 后面引号里的东西按原样输出。上面的例子可改写如下：

```
>>> d=r"c:\news"
>>> d
'c:\\news'
>>> print(d)
c:\news
```

```
>>>
```

2.2.9 range

Python 中的内置函数 range(n)表示一个从 0 到 n-1 的长度为 n 的序列。

当然，可以自定义我们需要的起始点和结束点，例如：

```
>>> range(1,5)  #代表从 1 到 5(不包含 5)，即 1、2、3、4
>>>
```

range(n)函数还可以定义步长。

下面我们定义一个从 1 开始到 30 结束，步长为 3 的列表：

```
>>> range(1,30,3)
range(1, 30, 3)
>>> list(range(1, 30, 3)) #这里用 list 列表把值显示出来，list 在后面介绍
[1, 4, 7, 10, 13, 16, 19, 22, 25, 28]
>>>
```

默认情况下，range()的起始值是 0。

在 numpy 模块中，arange()类似于 range 函数，调用时须导入该模块：

```
import numpy
numpy.arange(n)
```

2.2.10 元组、列表、字典、集合

1．tuple

元组(tuple)类似于向量，元组的元素不能修改。元组写在小括号里，元素之间用逗号隔开，和向量的写法一致。元组中的元素类型也可以不相同，示例如下：

```
>>> a = (1991, 2014, 'physics', 'math')
>>> print(a, type(a), len(a))
(1991, 2014, 'physics', 'math') <class 'tuple'> 4
>>>
```

元组与字符串类似，可以被索引且下标索引从 0 开始，也可以进行截取切片(其实，可以把字符串看作一种特殊的元组)。例如：

```
>>> tup = (1, 2, 3, 4, 5, 6)
>>> print(tup[0], tup[1:5])
1 (2, 3, 4, 5)
>>> tup[0] = 11            # 修改元组元素的操作是非法的
Traceback (most recent call last):
  File "<pyshell#21>", line 1, in <module>
```

```
    tup[0] = 11
TypeError: 'tuple' object does not support item assignment
>>>
```

虽然 tuple 的元素不可改变，但它可以包含可变的对象，比如 list 列表。对于构造包含 0 个或 1 个元素的 tuple 是个特殊的问题，所以有一些额外的语法规则：

```
>>> tup1 = ()   #空元组
>>> tup1
()
>>> tup2 = (20,) #创建只有一个元素的元组，该元素后面的逗号不可忽略
>>> tup2
(20,)
>>> tup3 = (20)
>>> tup3
20
>>>
```

注意： 元组(2)其实就是数字 2，仍然是整型，但(2,)就是元组。元组是不可添加和删除的。另外，元组也支持用“+”操作符。例如：

```
>>> tup1, tup2 = (1, 2, 3), (4, 5, 6)
>>> print(tup1+tup2)
(1, 2, 3, 4, 5, 6)
>>>
```

元组由不同的元素组成，每个元素可以存储不同类型的数据，而元组中的元素则代表不同的数据项。元组的创建可以不定长，但创建后和字符串一样，都是不可修改的。

例如：

```
>>> user=(1,2,3)
>>> user[0]=2

 Traceback (most recent call last):
   File "<pyshell5>", line 1, in <module>
     user[0]=2
 TypeError: 'tuple' object does not support item assignment
>>>
```

元组的添加：

```
>>> user = ('01','02','03','04')
>>> user = (user,'05')  #注意结果与两元组用“+”操作符结果的比较
>>> user
(('01', '02', '03', '04'), '05')
```

```
>>>
```

元组的访问：

```
>>> user = ('01','02','03','04')
>>>user[0]
 '01'
>>>user[2]
'03'
>>>
```

元组包含以下内置函数。

- len(tuple)：计算元组元素的个数。
- max(tuple)：返回元组中元素的最大值。
- min(tuple)：返回元组中元素的最小值。
- tuple(list)：将列表转换为元组(list 在后面介绍)。

注意： cmp()函数在 Python 3.5 中已经被删除。

二元元组(二维)的访问：

```
>>>user1 = (1,2,3)
>>>user2 = (4,5,6)
>>>user = (user1,user2)
>>>print(user[1][2])
   6
```

元组的解包：

```
>>> user = (1,2,3)
>>> a,b,c = user     #变量个数要等于元组的长度
>>> a
1
>>> b
2
>>> c
3
>>>
```

2．list

列表(list)用方括号[]标识，其元素写在方括号之间，并用逗号分隔开。列表中元素的类型可以不相同，例如 a = [“I”,“you”,“he”,5]。

列表的索引同字符串和元组一样，如图 2-1 所示。

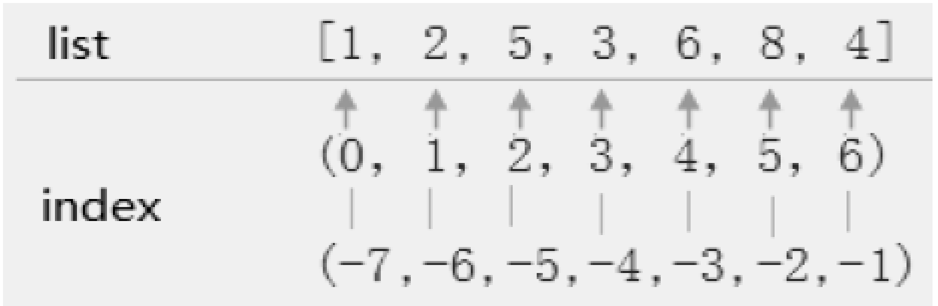


图 2-1　list 索引图

列表 array = [1, 2, 5, 3, 6, 8, 4]对应的元素 1=array[0]= array[-7]，以此类推。

从 0 开始，0 表示第一个元素索引，-1 表示最后一个元素索引，-len(array)表示第一个元素的索引，len(array)-1 表示最后一个元素的索引。

len(list)表示取 list 的元素数量，也即 list 的长度。

创建连续的 list：

```
>>> list(range(1,5))
[1, 2, 3, 4]
>>> list(range(1, 10, 2))  #步长为 2，从 1 开始，每隔 2 取一个数
[1, 3, 5, 7, 9]
>>>
```

和字符串一样，列表同样可以被索引和切片，列表被切片后，返回一个包含所需元素的新列表。例如：

```
>>> a = ["I","you","he",5]
>>> a[1:3]
['you', 'he']
>>> a[1]
'you'
>>> a[1]='she'
>>> a
['I', 'she', 'he', 5]
>>>
```

从上面的语句可以看出，列表是可以改变的，a[1]由原来的'you'变成了'she'。

列表还支持串联操作，使用“+”操作符：

```
>>> a = [1, 2, 3, 4, 5]
>>> a + [6, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8]
>>>
```

使用 a[n:n]=[q]可以在列表 a 中的 n 位置插入一个值 q。例如：

```
>>> a = [1, 2, 4, 5]
>>> a[2:2]=[3]   #在列表中的某位置插入一个值
```

```
>>> a
[1, 2, 3, 4, 5]
>>> len(a)        #测试列表的长度(含元素的个数)
5
>>>
```

使用 list(str)可以将字符串转化为列表。例如：

```
>>> word='hello'
>>> list(word)    #将字符串转化为列表
['h', 'e', 'l', 'l', 'o']
>>>
```

下面介绍 list 的有关方法。

(1) list 的元素追加。

① L.append(var)：追加元素，追加的元素可以是一个 list、数、字符串等。

例如：

```
>>> w=[1,2,'L']
>>> q=[8,9]
>>> w.append(q)    #不能写成 w=w.append(q)
>>> w
[1, 2, 'L', [8, 9]]
>>>
```

注意： append 方法不能返回值，所以像 w=w.append(q)这样写是错误的。

② L.extend(list)：合并两个列表，即把 list 追加到 L 列表中。不能追加单个元素。

例如：

```
>>> w=[1,2,'L']
>>> q=[8,9]
>>> w.extend(q)
>>> w
[1, 2, 'L', 8, 9]
>>>
```

注意： append 和 extend 的区别是：append 是对列表添加元素，extent 是合并两个列表。

③ L.insert(index,var)：在 index 位置插入 var 元素。

例如：

```
>>>a = ["I","you","he",5]
```

```
>>>a.insert(1,'love')
>>>a
 ['I', 'love', 'you', 'he', 5]
>>>
```

(2)　从 list 删除元素。

使用 L.pop(index)从 list 删除元素，返回被删除的 index 位置元素，只能删一个元素，并从 list 中删除这个元素。默认删除最后一个元素。例如：

```
>>>a = ["I","you","he",5]
>>>a.pop(2)
'he'
>>>a
['I', 'you', 5]
>>>a.pop()
5
>>>a
['I', 'you']
>>>
```

使用 del L[index]删除指定索引的元素。例如：

```
>>>a = ["I","you","he",5]
>>>del a[3]
>>>a
 ['I', 'you', 'he']
>>>
```

使用 del L[m:n]删除指定索引范围的元素。例如：

```
>>> a = [1, 2, 3, 4, 5]
>>> del a[1:3]     #删除 list 中索引为 1、2 的值
>>> a
[1, 4, 5]
>>>
```

使用 L.remove(var)删除第一次出现的 var 元素。例如：

```
>>>li = [1,2,3,4,5,4,6]
>>>li.remove(4)
>>>li
[1, 2, 3, 5, 4, 6]
>>>
```

另外，还可以使用切片的方法来删除。例如：

```
>>>li = [1,2,3,4,5,6]
```

```
>>>li = li[:-1]
>>>li
[1, 2, 3, 4, 5]
>>>
```

(3) list 操作符(:、+、*)。例如：

```
>>> a = ["I","you","he",5]
>>> a[1:]             #片段操作符，用于子 list 的提取
['you', 'he', 5]
>>> [1,2]+[3,4]        #同 extend()
[1, 2, 3, 4]
>>> [2]*4
[2, 2, 2, 2]
>>>
```

(4) list 的索引冒号用法。

冒号前后表示索引切片的起止位置。例如：

```
>>> array = [1, 2, 5, 3, 6, 8, 4]
>>> array[0:] #列出索引 0 以后的
[1, 2, 5, 3, 6, 8, 4]
>>> array[1:] #列出索引 1 以后的
[2, 5, 3, 6, 8, 4]
>>> array[:-1] #列出索引-1 之前的
[1, 2, 5, 3, 6, 8]
>>> array[3:-3] #列出索引 3 到索引-3 之间的
[3]
>>>
```

两个冒号[::]表示取全部索引。例如：

```
>>> array = [1, 2, 5, 3, 6, 8, 4]
>>> array[::2]  #表示步长为 2 取元素，即隔一个元素取一个元素
[1, 5, 6, 4]
>>> array[2::]
[5, 3, 6, 8, 4]
>>> array[::3]
[1, 3, 4]
>>> array[::4]
[1, 6]
```

双冒号[::]还可以形成 reverse 函数的效果。例如：

```
>>> array = [1, 2, 5, 3, 6, 8, 4]
>>> array[::-1]
```

```
[4, 8, 6, 3, 5, 2, 1]
>>> array[::-2]
[4, 6, 5, 1]
>>>
```

(5)　list 排序。

对列表进行排序可以使用 sort()和 sorted()函数。

①　sort()：此函数对列表排序时，会改变列表本身，从而让其中的元素按一定的顺序排列。例如：

```
>>> a = [3,2,5,4,9,8,1]
>>> a.sort()
>>> a
[1, 2, 3, 4, 5, 8, 9]
>>> sort(a)
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    sort(a)
NameError: name 'sort' is not defined
>>> help(a)
… … …
sort(...)
   L.sort(key=None, reverse=False) -> None -- stable sort *IN PLACE*
>>>
```

从 help 可知，sort()默认按从小到大排序，可以添加参数 reverse=True，变成从大到小排列。例如：

```
>>> e=[1, 3, 2, 4, 5]
>>> e.sort(reverse=True)
>>> e
[5, 4, 3, 2, 1]
>>>
```

注意： sort()函数使用的是“.”方法，即 a.sort()，使用 sort(a)就会报错。sort()函数会改变原来的列表，且函数返回值为空，即 None。因此，如果需要一个已排好序的列表副本，同时又要保留原有列表不被改变，就不能直接简单地使用 sort()函数(为了实现上述功能，可以使用 sorted(a)方法)。例如:

```
>>> L = [3,2,5,4,9,7,1]
>>> e = L.sort()              #返回空值 None
>>> print(e)
None
```

```
>>> L
[1, 2, 3, 4, 5, 7, 9]
>>>
```

② sorted()：此函数对列表进行排序时，直接获取列表排序的一个副本。sorted()函数可以用于任何可迭代的对象。例如：

```
>>> a = [3,2,5,4,9,8,1]
>>> sorted(a)                #对 a 排序后产生一个新的列表
[1, 2, 3, 4, 5, 8, 9]
>>> a
[3, 2, 5, 4, 9, 8, 1]
>>>
```

注意： ① a.sort()和 sorted(a)有区别：sorted(a)产生的是一个新列表，不改变原列表 a；而 a.sort()是对列表 a 直接排序，破坏了原列表。② sorted()既产生新的排序列表又保持原列表不被改变，这个功能也可以通过拷贝副本的方法来实现：先获取列表 a 的一个副本 b，然后再对 b 进行 b.sort()排序。

为了理解得更深刻，我们将对 a 和 b 的存储地址进行查验，代码如下：

```
>>> a = [3,2,5,4,9,8,1]
>>> id(a)                  #查看 a 的存储地址
55494776
>>> b=a[:]                 #拷贝一个副本 b
>>> b
[3, 2, 5, 4, 9, 8, 1]
>>> id(b)                           #查验 b 的存储地址
55486264                             #发现 b 和 a 地址不一致，说明复制了一份
>>> c=a                             #再复制一个副本 c，比较 c 和 a 的存储地址
>>> c
[3, 2, 5, 4, 9, 8, 1]
>>> id(c)                           #查验 c 的存储地址
55494776          #发现 c 的地址跟 a 一致，说明 c 是 a 的一个标签，不是真复制
>>> b.sort()                        #对 b 进行排序
>>> b
[1, 2, 3, 4, 5, 8, 9]
>>> a                               #b 排序后对 a 没有影响
[3, 2, 5, 4, 9, 8, 1]

>>> c.sort()                        #对 c 进行排序
>>> c
[1, 2, 3, 4, 5, 8, 9]
>>> a                               #c 排序后对 a 有影响
```

```
[1, 2, 3, 4, 5, 8, 9]
>>>
```

从上面代码显示的存储地址知道，c 仅仅是 a 的一个标签，并不是真正意义上的复制，不论是 a 改变，还是 c 改变，其实改变的都是同一个地址里的内容，所以互相有影响。只有 b 才是真正意义上的拷贝，后面我们还会遇到“深拷贝”。再如：

```
>>> x=[4,6,2,1,7,9,4]
>>> y=x[:]
>>> y.sort()
>>> print(x)
[4, 6, 2, 1, 7, 9, 4]
>>> print(y)
[1, 2, 4, 4, 6, 7, 9]
>>>
```

说明： 调用 x[:]得到的是包含了 x 所有元素的切片，这是一种很有效率的复制整个列表的方法。通过 y=x 简单地将 x 赋值给 y，仅仅是给 y“贴”了一个指向 x 的标签，最终 x 和 y 都指向了同一个列表。

③　reverse()：此函数用来进行倒序排列。例如：

```
>>> e=[1, 3, 2, 4, 5]
>>> e.reverse()
>>> e
[5, 4, 3, 2, 1]
>>>
```

或者：

```
>>> reversed([1,2,'L'])  #这样返回的是一个迭代器，可以用 list 转化为列表
<list_reverseiterator object at 0x02C921B0>
>>> list(reversed([1,2,'L']))
['L', 2, 1]
>>>
```

或者：

```
>>> w=[1,2,'L']
>>> w[::-1]
['L', 2, 1]
>>>
```

对字符串也可以同样反转。

(6) 其他方法。

- L.count(var)：返回 var 元素在列表中出现的个数。
- L.index(var)：返回第一个 var 元素的位置，无则抛出异常。

列表对象常用的方法汇总如表 2-5 所示。

表 2-5　list 对象常用的方法

list.append(x)	把一个元素添加到列表的结尾，相当于 a[len(a):] = [x]
list.extend(L)	将一个给定列表中的所有元素都添加到另一个列表中，相当于 a[len(a):] = L
list.insert(i, x)	在指定的索引位置 i 插入一个元素 x。如 a.insert(0, x)会插入到整个列表之前，而 a.insert(len(a), x)相当于 a.append(x)
list.remove(x)	删除列表中第一次出现的值为 x 的元素。如果没有这样的元素，就会返回一个错误
list.pop(i)	从列表的指定位置删除元素，并将其返回。如果没有指定索引，a.pop()返回最后一个元素，该元素随即从列表中被删除
list.index(x)	返回列表中第一个值为 x 的索引。如果没有匹配的元素，就会返回一个错误
list.count(x)	返回 x 在列表中出现的次数(可以用于做列表中的查重)
list.sort()	对列表中的元素就地进行排序(改变了原列表)
list.reverse()	就地倒排列表中的元素(改变了原列表)

3. dict

先来看列表存储通信录。

【例 2-4】通信录的存储：

```
>>> name=["Ben","Jone","Jhon","Jerry","Anny","Ivy","Jan","Wong"]
>>> tel=[6601,6602,6603,6604,6605,6606,6607,6608]
```

这里通信录存储在两个 list 中(一个 list 是姓名，一个 list 是对应的手机号)，但是要查阅某个人的电话号码，显得很不方便。

为了便于查阅，在 Python 中有另外一种存储方式：字典。

字典是一种映射类型(mapping type)，它是一个无序的“键: 值”对集合。每一个元素都是 pair，包含关键字 key、value 两部分。

key 是 Integer 或 string 类型，value 是任意类型。即：

```
{key: value}
```

关键字(key)必须使用不可变类型，在同一个字典中，关键字必须互不相同。例如：

```
>>> dic = {}    #创建一个空字典
>>> dic_tel = {'Jack':1557, 'Tom':1320, 'Rose':1886} #创建一个字典
>>> print(dic_tel)   #打印字典
```

```
{'Tom': 1320, 'Rose': 1886, 'Jack': 1557}
>>>
```

所以例 2-4 也可以用字典来存储。下面用遍历的方法来操作(遍历将在后面介绍)，其原理是：每次从 name 中取一个姓名，记为 n1，再从 tel 中取对应的号码，记为 t1，再把 n1 和 t1 组成键值对 n1:t1，作为字典 Tellbook 中的一个元素，如此循环，就全部构成了字典的元素。

【例 2-5】列表转成字典：

```
>>> name=["Ben","Jone","Jhon","Jerry","Anny","Ivy","Jan","Wong"]
>>> tel=[6601,6602,6603,6604,6605,6606,6607,6608]
>>> Tellbook={}  #创建一个空字典
>>> for i in range(len(name)):
    d1="{}".format(name[i])     #从 name 中取一个姓名
    d2="{}".format(tel[i])      #从 tel 中取一个电话
    Tellbook[d1]=d2             #再把 d2 赋值给字典 Tellbook 的 d1 键
>>> print(Tellbook)
{'Jan': '6607', 'Ben': '6601', 'Ivy': '6606', 'Anny': '6605',
'Wong': '6608', 'Jhon': '6603', 'Jone': '6602', 'Jerry': '6604'}
>>>
```

(1)　字典的增、删、改、查。

以下为字典的一些常用操作方法示例：

```
Tellbook['Wang'] = 3          #给键赋值，若键不存在，则直接创建此键
del Tellbook['Wong']          #删除一个键值对
Tellbook['Ben']               #通过 key 查询对应的值
list(Tellbook.keys())            #返回所有 key 组成的 list
list(Tellbook.values())          #返回所有 value 组成的 list
sorted(Tellbook.keys())          #按 key 对字典排序
'Ben' in Tellbook             #成员测试
'Mary' not in Tellbook          #成员测试
```

可以用构造函数 dict()直接从键值对构建字典，例如：

```
>>> dict([('sape', 4139), ('guido', 4127), ('jack', 4098)])
{'guido': 4127, 'jack': 4098, 'sape': 4139}
>>> dict(sape=4139, guido=4127, jack=4098)
{'guido': 4127, 'jack': 4098, 'sape': 4139}
>>>
```

字典有.items 方法：将字典里的元素(一个键值对)转化为元组，作为列表的一个元素。例如：

```
>>> d= {'a':1, 'b':2, 'c':3}
>>> t= d.items()
>>> print(t)
dict_items([('b', 2), ('c', 3), ('a', 1)])
>>> list(t)
[('b', 2), ('c', 3), ('a', 1)]
>>>
```

当然，上面的过程是可逆的，即元组列表可以初始化成字典：

```
>>> t=[('a',1),( 'b',2),( 'c',3)]
>>> d=dict(t)
>>> print(d)
{'b': 2, 'c': 3, 'a': 1}
>>>
```

使用 update 函数可以合并两个字典：

```
>>> dict = {'Name': 'Zara', 'Age': 7}
>>> dict2 = {'Sex': 'female'}
>>> dict.update(dict2)
>>> dict
{'Age': 7, 'Sex': 'female', 'Name': 'Zara'}
>>>
```

用 tuple 作为键可以创建字典：

```
>>> seq = ('name', 'age', 'sex')
>>> dict = dict.fromkeys(seq)  #给字典 key 的赋值来自 seq
>>> dict     #因为仅有 key，没有 value，所以显示键值为空 None
{'sex': None, 'age': None, 'name': None}
>>> dict = dict.fromkeys(seq, 10) #给字典键值对赋值，这里假设都赋 10
>>> dict
{'sex': 10, 'age': 10, 'name': 10}
>>>
```

字典有下列“.”方法：

```
D.get(key, 0)      #同 dict[key]，此处参数为 0(也可以是其他的，如 none)，
                   #表示字典 D 中若没有 key 键则返回指定的值 0
D.keys()           #返回字典键的列表
D.values()         #返回字典值的列表
D.items()          #将字典转化为元组作为元素的一个列表
D.update(dict2)      #合并字典，将 dict2 增加到当前的字典中
D.pop(key)         #从字典中删除指定的键值对，键名这个参数必须有
D.popitem()        #没有参数则从字典中随机删除一个键值对。已空则抛出异常
```

```
D.clear()          #清空字典，不同于 del dict 是删除字典
D.copy()           #拷贝字典
Dict1=dict.copy()    #克隆，即另一个浅拷贝，深拷贝则是 deepcopy
```

示例如下：

```
>>> dict={'sex': 10, 'age': 10, 'name': 10}
>>> dict.get('sex', 'None') #若没有 sex 键，则返回指定的 None
10
>>> dict.keys()      #要想获取键名列表，直接 list(dict.keys())即可
dict_keys(['sex', 'age', 'name'])
>>> dict.items()     #要想获取键值列表，直接 tuple(dict.keys())即可
dict_items([('sex', 10), ('age', 10), ('name', 10)])
>>> dict.pop('sex')     #删除 sex 键值对
10
>>> dict
{'age': 10, 'name': 10}
>>> dict['sex']=10    #增加键值对 sex: 10
>>> dict
{'sex': 10, 'age': 10, 'name': 10}
>>> dict.popitem()  #随机删除一个键值对
('sex', 10)
>>> dict
{'age': 10, 'name': 10}
>>> dict1=dict.copy()  #复制(浅拷贝)一个字典，浅拷贝只对简单类型拷贝
>>> dict1
{'age': 10, 'name': 10}
>>> dict.clear()  #清空字典，不是删除字典，即得到一个空字典
>>> dict          #dict 已经变成了一个空字典
{}
>>> import copy   #导入 copy 函数(或者是模块)
>>> dict=copy.deepcopy(dict1)   #深拷贝，将 dict1 拷贝给 dict
>>> dict
{'age': 10, 'name': 10}
>>>
```

【例 2-6】字典内置 get 方法的调用。

假设用户在终端输入字符串：“1”、“2”或“3”，则返回对应的内容，如果输入其他的，则返回“error”。程序如下：

```
>>>info = {'1':'first','2':'second','3':'third'}
>>>print(info.get(input('input type you number:'),'error'))
input type you number:2
second
```

```
>>>
```

回顾一下 D.get(key, None)的用法：表示字典 D 中若有 key 这个键，则返回其键值；若没有 key 键，则返回指定的值 None。当然，这里的 None 也可以改写成 D.get(key, ‘哈哈，别逗，你输错了！’)，当没有 key 键的时候，则返回‘哈哈，别逗，你输错了！’

具体程序如下：

```
>>>info = {'1':'first','2':'second','3':'third'}
>>>print(info.get(input('input type you number:'),'哈哈，别逗，你输错了！'))
input type you number:5
哈哈，别逗，你输错了！
>>>
```

(2) 字典的排序。

在程序中使用字典进行数据信息统计时，由于字典是无序的，所以打印输出的字典内容也是无序的。因此，为了方便结果查看，需要对字典进行排序。Python 中字典的排序分为按键 key 排序和按值 value 排序。

① 按“值”排序。

按“值”排序，就是根据字典的 value 进行排序，可以使用内置的 sorted()函数。

例如：

```
>>> dict={'班级': 1, 'age': 10, 'score': 10}
>>> sorted(dict.items(), key=lambda e:e[1], reverse=True)
[('score', 10), ('age', 10), ('班级', 1)]
>>>
```

其中 e 表示 dict.items()中的一个元素，e[1]则表示按值排序。reverse=False 可以省略，默认为升序排列。

说明： 字典的.items()函数返回的是一个列表，列表的每个元素都是一个键和值组成的元组。因此，sorted(dict.items(), key=lambda e:e[1], reverse=True)返回的值同样是由元组组成的列表。lambda 函数后面会专门介绍。

② 按“键”排序。

对字典进行按键排序也可以使用 sorted()函数，只要修改为 sorted(dict.items(), key=lambda e:e[0], reverse=True)即可。

4. set

集合(set)是一个无序、不重复元素的集，set 的基本功能是去重。可以使用大括号{}或者 set()函数创建 set 集合。

注意： 创建空集合必须用 set()，不能使用{ }，因为{ }是用来创建空字典的。

例如：

```
>>> student = {'Tom', 'Jim', 'Mary', 'Tom', 'Jack', 'Rose'}  #有重复元素
>>> print(student)       #重复的元素被自动去掉
{'Jim', 'Jack', 'Mary', 'Rose', 'Tom'}
>>> 'Rose' in student    #membership testing(成员测试)
True
>>>student.add('Ben')   #增加一个元素
>>> print(student)
>>>{'Jim', 'Jack', 'Mary', 'Rose', 'Tom', 'Ben'}
>>>
```

集合的运算：

```
>>> a = set('abracadabra')  #将字符串拆成集合
>>> a
{'b', 'd', 'r', 'a', 'c'}
>>> b = set('alacazam')      #将字符串拆成集合
>>> b
{'l','m','z','a','c'}
>>> a-b                       #从 a 中去除 b 的元素
{'b', 'd', 'r'}
>>> a|b                   #a 和 b 的并集
{'l','r','a','c','z','b','m','d'}
>>> a&b              #提取 a 和 b 的公共元素——交集
{'a', 'c'}
>>> a^b    #提取 a 和 b 中不同时存在的元素(交集的补集，也叫对称差)
{'l','r','z','m','b','d'}
>>>
```

集合的去重。集合有过滤重复元素的功能，自动将重复元素删除。例如：

```
>>> set((2,2,2,4,4))
{2, 4}
>>>
```

2.2.11　格式化输出

1．%格式化输出

Python 用 print 进行格式化输出，有以下几种模式。

(1)　输出字符串：

```
>>> print("His name is %s"%("Aviad"))
His name is Aviad
>>>
```

打印输出的内容里(“His name is %s”%(“Aviad”))有两个%，其中%s 表示先在“His name is %s”这个字符串中占个位置，而后面的“Aviad”才是%s 位置上真正要显示的内容，也就是%s 位置上要显示的内容在后面%的括号内，即“Aviad”。

(2) 输出整数：

```
>>> print("He is %d years old"%(25))
He is 25 years old
>>>
```

第一个例子中%s 是要替字符串占位置，这里的%d 表示要输出后面提供的整数。

(3) 输出浮点数：

```
>>> print("His height is %f m"%(1.83))
His height is 1.830000 m
>>>
```

这里%f 表示输出后面提供的浮点数。

(4) 输出浮点数(指定保留小数点位数)：

```
>>> print("His height is %.2f m"%(1.83))
His height is 1.83 m
>>>
```

这里的%.2f 表示只显示小数点后两位数字，也就是指定了保留小数点位数。

(5) 输出指定占位符的宽度：

```
>>> print("Name:%10s Age:%8d Height:%8.2f"%("Aviad",25,1.83))
Name:     Aviad Age:      25 Height:    1.83
>>> print('i love %.2s'%'python')
i love py
>>>
```

(6) 输出指定占位符的宽度(左对齐)：

```
>>> print("Name:%-10s Age:%-8d Height:%-8.2f"%("Aviad",25,1.83))
Name:Aviad     Age:25      Height:1.83
>>>
```

(7) 指定占位符(0 或者空格)：

```
>>> print("Name:%-10s Age:%08d Height:%08.2f"%("Aviad",25,1.83))
Name:Aviad     Age:00000025 Height:00001.83
```

```
>>>
```

2．format 格式化输出

格式化字符串的函数 str.format()可谓威力十足。format 函数跟前面的%型格式化字符串相比，其优越性是通过{}和.来代替%，看如下示例：

```
>>> '{0},{1}'.format('yubg',39)        #这里的 0 和 1 表示的是位置索引
'yubg,39'
>>> '{},{}'.format('yubg',39)          #位置索引也可以为空
'yubg,39'
>>> '{1},{0},{1}'.format('yubg',39)  #可以接受多个参数，位置可以无序
'39,yubg,39'
>>>
```

format 的关键字参数：

```
>>> '{name},{age}'.format(age=39,name='yubg')
'yubg,39'
>>>
```

有多个输出需要多个占位符时，可以通过设定下标加以区分：

```
>>> p=['yubg',39]
>>> '{0[0]},{0[1]}'.format(p)
'yubg,39'
>>>
```

格式限定符：它有着丰富的格式，比如填充与对齐，语法为{}中带:号。填充和对齐经常一起使用。

- ^——居中，后面带宽度。
- <——左对齐，后面带宽度。
- >——右对齐，后面带宽度。
- :——后面带填充的字符，只能是一个字符，不指定时，默认是用空格填充。

例如：

```
>>> '{:>8}'.format('189') #默认是用空格来占位，要显示的内容靠右对齐
'     189'
>>> '{:0>8}'.format('189') #用 0 来占位
'00000189'
>>> '{:a<8}'.format('189') #用字母 a 来占位，要显示的内容靠左对齐
'189aaaaa'
>>> '{:*^7}'.format('189')#用*来占位，共显示 7 位，要显示的内容居中
'**189**'
```

```
>>>
```

精度与类型 f：精度常跟浮点数类型 f 一起使用。

例如：

```
>>> '{:.2f}'.format(321.33345)    #保留两位有效数字
'321.33'
>>>
```

2.2.12　strip、split

1．strip()

strip、lstrip、rstrip 的使用方法如下。

- strip：去掉字符串两边的空格。
- lstrip：去掉字符串左边的空格。
- rstrip：去掉字符串右边的空格。

Python 中的 strip 用于去除字符串的首尾字符，同理，lstrip 用于去除左边的字符，rstrip 用于去除右边的字符。这三个函数都可以传入一个参数，指定要去除的首尾字符。需要注意的是，传入的是一个字符数组时，编译器去除两端所有相应的字符，直到没有匹配的字符为止，例如：

```
>>> theString = 'saaaay yes or no yaaaass'
>>> print(theString.strip('say'))
 yes or no
>>> theString
'saaaay yes or no yaaaass'
>>>
```

theString 依次被去除首尾在[‘s’,‘a’,‘y’]列表内的字符，直到字符不在数组内，所以，输出的结果为：yes or no。当然，这里生成的是一个“副本”，不会改变原来的字符串 theString。

lstrip 和 rstrip 的原理一样。当没有传入参数时，是默认去除首尾的空格。

例如：

```
>>> theString = 'saaaay yes or no yaaaass'
>>> print(theString.strip('say') )
 yes or no
>>> theString.strip('say')
' yes or no '
```

```
>>> theString.strip('say ')
'es or no'
>>> theString.lstrip('say')
' yes or no yaaaass'
>>> theString.rstrip('say')
'saaaay yes or no '
>>>
```

2．split()

split()的作用是对字符串进行分割。

(1)　按某个字符分割。如按‘.’进行分割：

```
>>> str = ('www.i-nuc.com')
>>> print(str)
www.i-nuc.com
>>> str_split = str.split('.')
>>> print(str_split)
['www', 'i-nuc', 'com']
>>>
```

(2)　按某个字符分割，且分割 n 次。如按‘.’分割 1 次：

```
>>> str = ('www.i-nuc.com')
>>> str_split = str.split('.',1)
>>> print(str_split)
['www', 'i-nuc.com']
>>>
```

(3)　按某个字符(或字符串)分割，且分割 n 次，并将分割完成的字符串(或字符)赋给新的 n+1 个变量。

如按‘.’分割字符，且分割 1 次，并将分割后的字符串赋给两个变量 str1 和 str2：

```
>>> url = ('www.i-nuc.com')
>>> str1, str2 = url.split('.', 1)
>>> print(str1,str2)
www i-nuc.com
>>> print(str1)
www
>>> print(str2)
i-nuc.com
>>>
```

【例 2-7】提取下面字符串中的 50,0,51：

```
>>> str="xxxxxxxxxxxx5 [50,0,51]>,xxxxxxxxxx"
>>> lst = str.split("[")[1].split("]")[0].split(",")
>>> print(lst)
['50', '0', '51']
>>>
```

分解如下：

```
>>> list =str.split("[")  #按照左边分割
>>> print(list)
['xxxxxxxxxxxx5 ', '50,0,51]>,xxxxxxxxxx']
>>> str.split("[")[1].split("]")  #再对 list 的 index=1 的元素按"]"分割
['50,0,51', '>,xxxxxxxxxx']
>>> str.split("[")[1].split("]")[0]  #提取分割后的第一个元素，即 index=0
'50,0,51'
>>> str.split("[")[1].split("]")[0].split(",")  #对提取后的按“,”分割
['50', '0', '51']
>>>
```

2.2.13 divmod()

divmod()：返回的商和余数是一个 tuple。

格式：divmod(被除数，除数)

返回值：(商，余数)

例如：

```
>>> t = divmod(7,3)
>>> t
(2, 1)
>>>
```

或者：

```
>>> quot,rem = divmod(7,3)
>>> print(quot,rem)
2 1
>>>
```

2.2.14 join()

使用 join()函数，可以把一个 list 或者 tuple 中所有的元素按照定义的分隔符(sep)连接起来，但限于元素是字符型。

语法：‘sep’.join(seq)

例如：

```
>>> a = ['a','b','c']
>>> sep = '|'
>>> x="|".join(a)
>>> x
'a|b|c'
>>>

>>> s=('1','2','3')
>>> "|".join(s)
'1|2|3'
>>> c=[1,2,3]
>>> '|'.join(c)
Traceback (most recent call last):
  File "<pyshell#50>", line 1, in <module>
    '|'.join(c)
TypeError: sequence item 0: expected str instance, int found
>>> c=['1','2','3']
>>> '|'.join(c)
'1|2|3'
>>>
```

本 章 小 结

本章知识点较多，重点是 list、tuple、dict、set。

(1)　测试变量类型：

```
type(变量)
```

(2)　转换变量类型：

```
str(变量)    #将变量转换为 str
int(变量)    #将变量转换为 int
```

(3)　查询相关命令的属性和方法 dir()：

```
>>> dir(list)
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__',
'__dir__', '__doc__', '__eq__', '__format__', '__ge__',
'__getattribute__', '__getitem__', '__gt__', '__hash__', '__iadd__',
'__imul__', '__init__', '__iter__', '__le__', '__len__', '__lt__',
```

```
'__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__',
'__repr__', '__reversed__', '__rmul__', '__setattr__', '__setitem__',
'__sizeof__', '__str__', '__subclasshook__', 'append', 'clear', 'copy',
'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
>>>
```

从上面的列表中可以看出，list 删除有两个属性 pop 和 remove(pop 默认删除最后一个元素，remove 删除首次出现的指定元素)。

(4) 查询已安装的模块：

```
help('modules')
```

对于初学者而言，也许 dir()和 help()这两个函数是最有用的，使用 dir()可以查看指定模块中包含的所有成员或者指定对象类型支持的操作，而 help()函数则返回指定模块或函数的说明文档。例如：list 和 tuple 是否都有 pop 和 sort 方法呢？那用 help 查一下，就很清楚了，并且列出了具体的用法：

```
>>> help(list)
Help on class list in module builtins:
 |  ...
 |  append(...)
 |   L.append(object)-> None -- append object to end
 |  pop(...)
 |   L.pop([index])->item--remove and return item at index (default last).
 |     Raises IndexError if list is empty or index is out of range.
 |  sort(...)
 |     L.sort(key=None, reverse=False) -> None -- stable sort *IN PLACE*
...
>>> help(tuple)
Help on class tuple in module builtins:
...
 | count(...)
 |   T.count(value) -> integer -- return number of occurrences of value
 |
 | index(...)
 |   T.index(value, [start, [stop]])->integer--return first index of
value.
 |     Raises ValueError if the value is not present.
>>>
```

(5) 查询两个变量的存储地址是否一致，使用 id()即可。

(6) 查询字符的 ASCII 码(十进制的)：

```
>>> ord('a')
```

```
97
>>>
```

反过来，有了十进制的整数，如何找出对应的字符？

```
>>> chr(97)
'a'
>>>
```

(7)　查找字符串的长度：

```
len()
```

(8)　str 通过索引能找出对应的元素，反过来，能否通过元素找出索引？

```
>>>s='python good'
>>>s[1]
'y'
>>>s.index('y')
1
>>>
```

(9)　tuple、list、string 的相同点。

每一个元素都可以通过索引来读取，都可以用 len 测长度，都可以使用加法“+”和数乘“*”。数乘表示将 tuple、list、string 重复数倍。

list 的.append、.insert、.pop、.del 和 list[n]赋值等方法属性均不能用于 tuple 和 str。

(10) str.split()是将字符型转化成 list，如下例：

```
>>> s='I love python，and\nyou\t?hehe'
>>> print(s)
I love python，and
you ?hehe
>>> s.split(",")    #英文“,”
['I love python，and\nyou\t?hehe']
>>> s.split("，")   #中文“，”
['I love python', 'and\nyou\t?hehe']
>>>
```

当分隔符不在字符串中时，会整体转化成一个 list。例如：

```
>>> s.split()
['I', 'love', 'python，and', 'you', '?hehe']
>>>
```

当分隔符省略时，会按所有的分隔符号分割，包括\n(换行)和\t(tab 缩进)等。

(11) Split 的逆运算：join。例如：

```
'sep'.join(list)
```

(12) 列表和元组之间是可以相互转化的：list(tuple)、tuple(list)。

元组的操作速度比列表快；列表可以改变，元组不可变，可以将列表转化为元组“写保护”；字典的 key 也要求不可变，所以元组可以作为字典的 key，但元素不能有重复。

(13) 字符串检测开头和结尾：string.endswith('str')、string.startswith('str')。

例如：

```
>>> file = 'F:\\data\\catering_dish_profit.xls'
>>> file.endswith('xls')      #判断 file 是否以 xls 结尾
True
>>>
>>> url = 'http://www.i-nuc.com'
>>> url.startswith('https') #判断 url 是否以 https 开头
False
>>>
```

(14) S.replace(被查找词，替换词)：查找与替换。例如：

```
>>> S='I love python, do you love python?'
>>> S.replace('python','R')
'I love R, do you love R?'
>>>
```

(15) re.sub(被替词，替换词，替换域，flags=re.IGNORECASE)：查找与替换，忽略大小写。例如：

```
>>> import re      #导入正则模块
>>> S='I love Python, do you love python?'
>>> re.sub('python','R',S)    #在 S 中用 R 替换 python
'I love Python, do you love R?'
>>> re.sub('python','R',S, flags=re.IGNORECASE) #替换时忽略大小写
'I love R, do you love R?'
>>> re.sub('python','R',S[0:15], flags=re.IGNORECASE)
'I love R, '
>>>
```

(16) Python 命名规范。

① 包名、模块名、局部变量名、函数名：全小写+下划线式驼峰。

如：this_is_var。

② 全局变量：全大写+下划线式驼峰。

如：GLOBAL_VAR。

③　类名：首字母大写式驼峰。

如：ClassName()。

④　关于下划线。

以单下划线开头，是弱内部使用标识，from M import *时，将不会导入该对象。

以双下划线开头的变量名，主要用于类内部标识类私有，不能直接访问。

双下划线开头且双下划线结尾的命名方法尽量不要用，这是标识。

练　　习

(1)　已知：a = 2, b = 3，要求：将 a 和 b 的值调换，并打印结果。

(2)　已知：a = 250, b = ‘250’，要求：阐述 a 和 b 所引用的对象的区别。

(3)　计算：100 除以 3 得到的商、余数分别是多少？如果保留 3 为小数，则结果将是多少？

(4)　请解释如下现象：

```
>>> round(2.675, 2)
2.67
>>>
```

(5)　精确计算：2 除以 6。要求：结果以分数形式(1/3)输出。

(6)　要求：在 print()里面将“明月几时有”和“把酒问青天”两句分两行输入，但输出结果时在一行。

(7)　编写程序，要求输入姓名和年龄，并且将年龄加 10 之后，与姓名一起输出。

(8)　将字符串“map”的字符顺序倒转为“pam”。

(9)　让用户输入一个单词，并显示这个单词的长度。

(10) 已知字符串：“Python is a widely used high-level, general-purpose, interpreted, dynamic programming language.” 要求：将字符串中每个单词的第一个字母都变成大写字母，最终样式如下：

```
'Python Is A Widely Used High-level, General-purpose, Interpreted,
Dynamic Programming Language.'
```

(11) 已知列表：[“python”, “java”, “c”, “c++”, “lisp”]。要求：用切片方式将此列表中的第 1、3、5 项取出来。

(12) 生成一个由 100 以内能够被 5 整除的数组成的列表，然后将该列表的数字从大到小排序。

(13) 已知两个列表：citys = [“suzhou”, “shanghai”, “hangzhou”, “nanjing”]，codes = [“0512”, “021”, “0571”, “025”]。要求：创建一个字典，以 citys 中的元素为 key，以 codes 中的元素为 value。

(14) 已知：列表 lst1=[1,2,3,4,5,6], lst2=[“a”,“b”,“c”,“d”]。要求：以 lst1 的元素为 key，以 lst2 的元素为 value 建立一个字典，并打印输出。

第 3 章

流程控制及函数与类

从本章开始，不再一行一行地执行代码，而是将整段代码写完后再执行。本章暂时还使用形态便捷的 Python 3.5。

从开始菜单里打开 Python 的 IDLE，在弹出的 Shell 窗口中选择 File → New File 菜单命令，即弹出一个空白的文本框窗口，如图 3-1 所示。

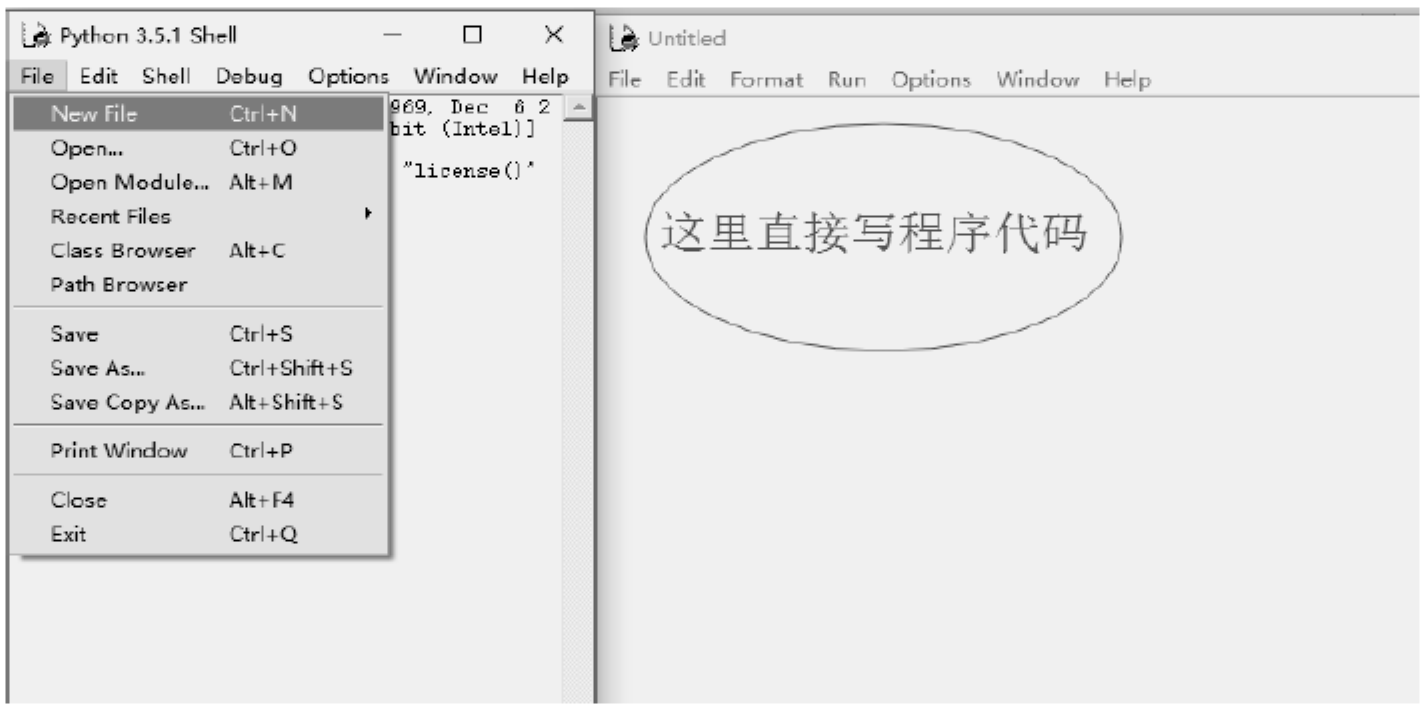


图 3-1　在 IDLE 中新建文件

利用第 2 章中的例 2-5 字典案例，一次性地把代码写完，并保存在指定的目录下，命名为 test1.py，后缀默认是.py。然后选择菜单栏中的 Run → Run Module 命令即可，或者直接按下快捷键 F5，即会弹出运行结果窗口。具体情况如图 3-2 所示。

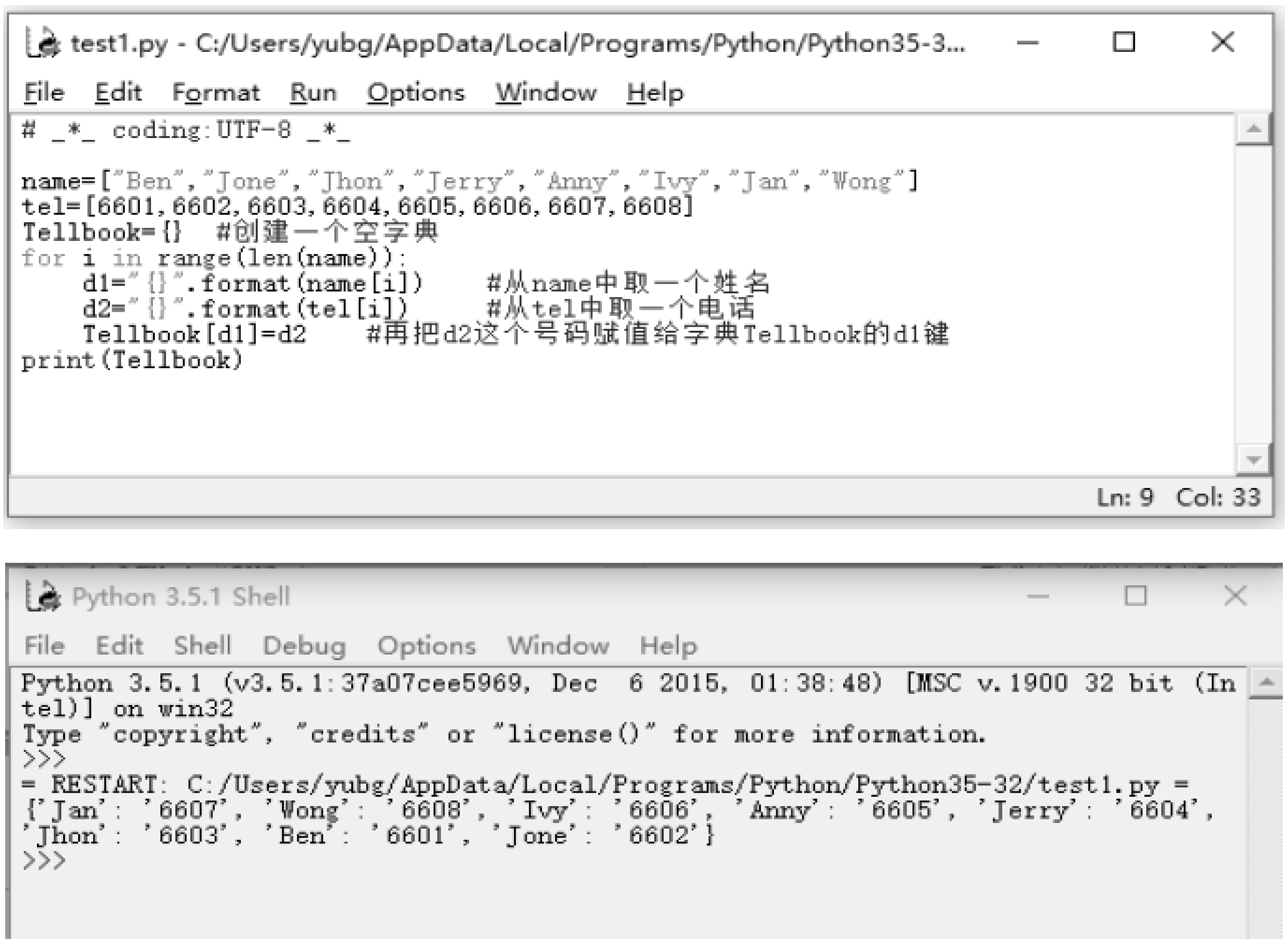


图 3-2　IDLE 新建文件界面及程序运行结果

当写多行测试代码时，会经常需要注释掉大段代码，这时候不再是每行添加#，而是在注释的代码段第一行前添加一个空行，写上三个单引号(''')或者三个双引号；再在段尾添加一行，也写上三个单引号或者三个双引号，也就是将要注释的代码段放在两个三引号

之间。当然，也可以将要注释的代码段选中，按下 F3 键即可；当要取消注释时，只须选中要取消注释的代码段，按下 F4 键即可。

注意： 在 Python 中，可以在各种编码间相互转换，有时因为编码的问题，会出现乱码。如果在“.py”文件中使用了中文，则需要在文件的第一行使用如下语句指定字符编码集：`# _*_ coding:UTF-8 _*_`。

关于代码操作快捷键，应牢记如下操作规则。

(1) 当有多行代码需要整体缩进时，选中代码，缩进用 Ctrl+]，取消缩进用 Ctrl+[。

(2) 注释多行代码除了使用三引号“包起来”的办法，还可以选中要注释的多行代码，按下 Alt+3，取消注释则按下 Alt+4。

常用的快捷键见表 3-1。

表 3-1　常用的快捷键

快 捷 键	功能说明
Alt+P	浏览上一条命令
Alt+N	浏览下一条命令
Ctrl+F6	重启 Shell
Alt+/	自动补全曾出现过的命令单词
Ctrl+]	缩进代码块
Ctrl+[	取消代码块缩进
Alt+3	注释代码块
Alt+4	取消代码块注释

当 Python 的 IDLE 中代码字体偏小时，可以在菜单栏中选择 Options → Configure IDLE 命令，在弹出的设置对话框中设置字体和大小，如图 3-3 所示。

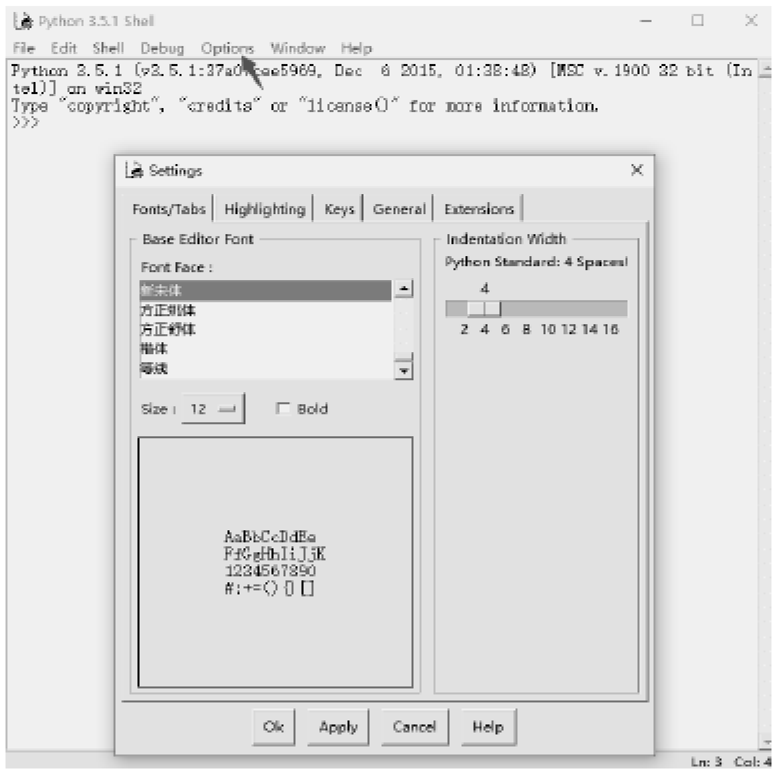


图 3-3　IDLE 界面参数设置

3.1 流程控制

3.1.1 if-else

if-else 分支语句结构的特点是当条件 condition 满足时，执行 if 下的语句块，当条件 condition 不满足时，执行 else 下的语句块。if-else 的语法结构如下：

```
if conditon:
    statement1
else:
    statement2
```

if-else 需要注意的格式问题：

- 在 if 和 else 行尾均要加冒号“:”。
- if 和 else 下的每条语句都要用缩进，即显示逻辑层次关系，缩进 4 个空格或者一个 Tab 键，Tab 键和空格不能混用，以免程序出错。
- 每个语句各占一行。

if-else 条件语句具有多种结构形式。

(1) 只有一个条件和单一的可能性操作时，如果满足条件就执行第二行。例如：

```
i=3
if i>1:
    print('you are right!')
```

(2) 当根据单一条件有两种不同操作时，可以使用 else，当条件满足时执行 if 下的语句块，条件不满足时执行 else 下的语句块，else 要与 if 对齐。例如：

```
i=3   #or i=0
if i>1:
    print('i 大于 1!')
else:
    print('i 不大于 1.')
```

(3) 如果还有更多的条件，可以使用 elif。例如：

```
if i>1:
    print('i 大于 1!')
elif i==1:
    print('i 等于 1!')
else:
    print('i 小于 1，或者说 i 不大于 1! ')
```

3.1.2　for 循环

使用 for var in field:，可以枚举 field 中的所有元素 var，并进行循环处理。

(1)　for 循环语句常用于遍历列表，基本句型如下：

```
>>> for i in [2,3,4]:
        print(i,end=',')

2,3,4,
>>>
```

上面的代码意思是将列表[2,3,4]中的每一个元素输出。在 print 语句中加上 end=‘,’ 表示将结果显示在一行中，用逗号(,)分隔开。

(2)　for 循环可以帮助处理字符串。假如想分别输出字符串中的每一个字母，则：

```
>>> for i in 'abc':
        print(i)

a
b
c
>>>
```

(3)　for 经常和 range 内置函数配合在一起使用。例如：

```
>>> for i in range(3):
        print(i)

0
1
2
>>>
```

(4)　for 循环可以用来生成列表。如用 range(3)来生成列表[0,1,2]，然后使用循环来计算 x 的平方，放入列表中：

```
>>> [x**2 for x in range(3)]
[0, 1, 4]
>>>
```

(5)　for 与 if 同用，表示按条件来生成列表。如生成 100 以内偶数平方构成的列表：

```
>>> [x**2 for x in range(10) if x%2==0]
[0, 4, 16, 36, 64]
```

```
>>>
```

3.1.3 while 循环

while 语句用于循环执行程序，即在某个条件下循环执行某段程序，以处理需要重复处理的相同任务，直到不满足循环条件时终止。

其基本形式如下：

```
while <判断条件>:
    <执行语句>
```

执行语句可以是单个语句或语句块。判断条件可以是任何表达式，任何非零或非空(null)的值均为 True。当判断条件为假 False 时，循环结束。

【例 3-1】在 Python 的 IDLE 编辑器中写入以下代码：

```
i=0
while i<5:
    print('This is '+str(i))     #将 i 转化为字符型才能用“+”连接输出
    i+=1                         #相当于 i=i+1
```

将以上代码保存，并按 F5 键运行，输出结果如下：

```
This is 0
This is 1
This is 2
This is 3
This is 4
This is 5
>>>
```

与 while 语句相关的还有另外两个重要的命令，即 continue 和 break，用来跳过循环。continue 用于跳过该次循环继续执行下一个循环，而 break 则是完全退出 while 循环。此外，“判断条件”还可以是个常数值，表示循环必定成立。

3.1.4 continue 和 break

在循环执行过程中，如果遇到 continue，则跳过这一次执行，继续进行下一次的循环操作。如共有 10 次循环，运行到第 8 次时遇到了 continue，那么第 8 次循环马上中止，后面的代码不再执行，继续开始第 9 次循环。

在循环执行过程中，只要遇到 break，就停止循环，跳出整个 while 循环。

【例 3-2】在 Python 的 IDLE 编辑器中写入以下代码：

```
# continue 和 break 的用法
# 1. 将小于 10 的偶数输出
i = 1
while i < 10:
    i += 1
    if i%2 > 0:    # 非偶数时跳过输出
        continue
    print(i)     # 输出偶数 2、4、6、8、10

# 2. 输出小于等于 10 的正整数
j = 1
while 1:       # 循环条件为 1 必定成立
    print(j)      # 输出 1~10
    j += 1
    if j > 10:    # 当 j 大于 10 时跳出循环
        break
```

输出结果如下：

```
2
4
6
8
10

1
2
3
4
5
6
7
8
9
10
>>>
```

在使用 continue 编写循环语句时，要避免误入死循环，如：

```
>>> i=1
>>> while i<10:
        if i%2 == 0:
            continue
        print(i)
        i+=1
```

本例欲输出小于 10 的奇数，但进入了死循环。因为当 i=2 时被整除，于是进入 continue，后面的 print(i)和 i+=1 都不再执行，但此时的 i 依然是等于 2，所以继续进入 i=2，如此循环往复，只有强行终止才能退出。

对上面的代码稍做修改即可正常运行了：

```
>>> i=1
>>> while i<10:
    if i%2 == 0:
        i+=1
        continue
    print(i)
    i+=1

1
3
5
7
9
>>>
```

3.2 遍　　历

如果给定一个 list 或 tuple，我们可以通过 for 循环来输出 list 或 tuple 中的每一个元素，这个过程称为遍历。这种遍历也称为迭代(Iteration)。在 Python 中，迭代是通过 for in 来完成的。

3.2.1 range()函数

可以用 range()函数生成一个列表：

```
>>> for i in range(5):
        print(i,end=',')

0,1,2,3,4,
>>>
```

range(5)中的参数 5 代表从 0 到 4 的一个长度为 5 的序列。Python 中的索引序列一般都是左闭右开的，即不包含右边的数据。此遍历也可以改写成一句：

```
>>> print([i for i in range(5)])
[0, 1, 2, 3, 4]
```

```
>>>
```

当然，可以自定义需要的起始点和结束点：

```
>>> list(range(1,5)) #代表从 1 到 5，不包含 5
[1, 2, 3, 4]
>>>
```

说明： 在以前的版本中，range()函数直接返回一个列表，但在 3.5 版本中，range()作为一个容器存在。当需要将它作为列表时，只要用 list 转化一下即可；如果需要将它作为元组，只要用 tuple 转化一下即可。例如:

```
>>> a=range(5)
>>> list(a)
[0, 1, 2, 3, 4]
>>> tuple(a)
(0, 1, 2, 3, 4)
>>>
```

定义一个从 5 开始到 100 结束的列表：

```
>>> listB =[i for i in range(5,100)]
>>> print(listB)
[5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77,
78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95,
96, 97, 98, 99]
>>>
```

range()函数还可以定义步长，下面定义一个从 1 开始到 30 结束，步长为 3 的列表：

```
>>> print('range(1,30,3)表示：',list(range(1,30,3)))
range(1,30,3)表示： [1, 4, 7, 10, 13, 16, 19, 22, 25, 28]
>>> listC =[i for i in range(1,30,3)]
>>> print(listC)
[1, 4, 7, 10, 13, 16, 19, 22, 25, 28]
```

range()函数也可以倒序，格式为 range(a,b,-1)，a 要大于 b。例如：

```
>>>list(range(5, 0, -1))
[5, 4, 3, 2, 1]
>>>list(range(5, 0, -2))  #步长为-2
 [5, 3, 1]
>>>
```

【例 3-3】对 array 进行排序，在 Python 的 IDLE 中输入下面的代码：

```
array = [1, 2, 5, 3, 6, 7, 4]
for i in range(len(array) - 1, 0, -1):
    print(i)
    for j in range(0, i):
       print(j ,end=',')
       if array[j] > array[j + 1]:
          array[j], array[j + 1] = array[j + 1], array[j]
print(array)
```

分析这段代码：

行 1：array = [1, 2, 5, 3, 6, 7, 4]，是一个乱序的 list。

行 2：for i in range(len(array) - 1, 0, -1):，替换后就成为 range(6,0,-1)，意思是从 6 到 0，步长为-1，随后把这些值循环赋给 i，i 的值将会是 6, 5, 4, 3, 2, 1。

行 4：for j in range(0, i)，这是一个循环赋值给 j 的语句，j 的值将会是[0, 1, 2, 3, 4, 5][0, 1, 2, 3, 4][0, 1, 2, 3][0, 1, 2][0, 1]。

那么上边两个循环嵌套起来将会是：

```
i=6
j: 0,1,2,3,4,5
i=5
j: 0,1,2,3,4
i=4
j: 0,1,2,3
i=3
j: 0,1,2
i=2
j: 0,1
```

行 6：if array[j] > array[j + 1]，判断 array 的前后两个元素大小。

行 7：array[j], array[j + 1] = array[j + 1], array[j]，替换赋值。

本例执行的结果如下：

```
6
0,1,2,3,4,5,5
0,1,2,3,4,4
0,1,2,3,3
0,1,2,2
0,1,1
0,[1, 2, 3, 4, 5, 6, 7]
```

其实，使用 sort()函数就能完成以上排序问题：

```
>>> array = [1, 2, 5, 3, 6, 7, 4]
>>> array.sort()
>>> array
[1, 2, 3, 4, 5, 6, 7]
>>>
```

【例 3-4】求 100 到 1000 的水仙花数。若 i=a^3+b^3+c^3，则称 i 为水仙花数。

```
>>> for i in range(100,1000):
        ge = i % 10
        shi= i //10 %10
        bai= i // 100
        if ge**3+shi**3+bai**3 == i:
            print(i)

153
370
371
407
>>>
```

3.2.2 列表与元组的遍历

如果需要遍历一个数字序列，可以使用 Python 中内建的函数 range()。

例如，下面遍历一个列表 test_list：

```
>>> test_list =[1,3,4,'Hongten',3,6,23,'hello',2]
>>> for i in range(len(test_list)):
         print(test_list[i],end=',')

1,3,4,Hongten,3,6,23,hello,2,
>>>
```

元组的遍历：

```
>>> tup=('abd','123')
>>> for i in range(len(tup)):
        print(tup[i])

abd
123
>>>
```

二元元组的遍历：

```
for i in range(len(user)):
    for j in range(len(user)):
        print('user['+str(i)+']['+str(j)+']=',user[i][j])
```

再如：

```
user=((1,2),(3,4))
for i in range(len(user)):
     for j in range(len(user)):
          print('user['+str(i)+']['+str(j)+']=',user[i][j])
```

输出：

```
user[0][0]= 1
user[0][1]= 2
user[1][0]= 3
user[1][1]= 4
```

多维元组需要稍微改动：

```
for i in range(len(user)):
    for j in range(len(user[i])):
       print('user['+str(i)+']['+str(j)+']=',user[i][j])
```

例如：

```
user=((1,2,3,4),(3,4,5,6),(5,6))
for i in range(len(user)):
     for j in range(len(user[i])):
          print('user['+str(i)+']['+str(j)+']=',user[i][j])
```

输出：

```
user[0][0]= 1
user[0][1]= 2
user[0][2]= 3
user[0][3]= 4
user[1][0]= 3
user[1][1]= 4
user[1][2]= 5
user[1][3]= 6
user[2][0]= 5
user[2][1]= 6
```

3.3 函　　数

3.3.1　函数的定义

在 Python 中，函数定义的基本形式如下：

```
def function(params):
    block
    return expression/value
```

说明如下。

(1)　在 Python 中采用 def 关键字进行函数的定义，不用指定返回值的类型，另外注意 def 行尾的冒号“:”不能丢；函数的命名一般首字母不要大写，以区别后面讲到的类。

(2)　函数参数 params 可以是零个、一个或者多个，同样，函数参数也不用指定参数类型，因为在 Python 中，变量都是弱类型，Python 会自动根据值来维护其类型。

(3)　return 语句是可选的，它可以在函数体内的任何地方出现，表示函数调用执行到此结束；如果没有 return 语句，会自动返回 NONE，如果有 return 语句，但 return 后面没有接表达式或者值，则也返回 NONE。返回的值就是输出的功能，return 可以返回多个值，如 return a,b,c，以逗号分隔，相当于返回一个 tuple(定值表)，相当于 return (a,b,c)。

应注意，函数体内部的语句在执行时，一旦执行到 return 时，函数就执行完毕，并将结果返回。

因此，函数内部通过条件判断和循环可以实现非常复杂的逻辑。如果没有 return 语句，函数执行完毕后也会返回结果，只是结果为 None。return None 可以简写为 return。

(4)　一般在 block 中还包含有一个注释体——函数文档，功能是解释这个函数的功用，用两个三引号包围起来，放在 block 的最前面，也是为了方便能够用 help 函数查询。

【例 3-5】在 Python 的 IDLE 中输入以下代码：

```
def printHello():
   print('Hello')

def readNum():
   for i in range(0,5):
      print(i)
   return

def add(a,b):
   return a+b

print(printHello())
print(readNum())
```

```
print(add(1,2))
```

输出结果如下：

```
>>>
Hello
None

0
1
2
3
4
None

3
>>>
```

3.3.2　函数的使用

在 Python 中，函数的使用有严格的规定，函数不允许前向引用，即函数应当定义在前，使用在后。

例如：

```
print(add(1,2))
def add(a,b):
    return a+b
```

程序的执行结果如下：

```
>>>
Traceback (most recent call last):
  File "C:/Users/yubg/AppData/Local/Programs/Python/Python35-32/test5.py",
line 1, in <module>
   print(add(1,2))
NameError: name 'add' is not defined
>>>
```

从报错中可以看出，命名为 add 的函数未进行定义。所以在调用某个函数时，必须确保此函数定义在调用之前，即先定义函数，然后再调用。上述程序可修改如下：

```
def add(a,b):
   return a+b
print(add(1,2))
```

运行结果如下：

```
>>>
3
>>>
```

3.3.3 形参和实参

形参全称是形式参数，在用 def 定义函数时，函数名后面括号里的变量称作形式参数。在调用函数时提供的值或者变量称作实际参数，实际参数简称为实参。

【例 3-6】形参和实参：

```
#这里的 a 和 b 就是形参
def add(a,b):
    return a+b

#这里的 1 和 2 是实参
add(1,2)

#这里的 x 和 y 是实参
x=2
y=3
add(x,y)
```

3.3.4 参数的传递和改变

在大多数高级语言中，对参数传递方式的理解一直是个难点和重点，因为它理解起来并不是那么直观明了。下面我们来探讨一下 Python 中函数的参数传递问题。

在讨论此问题之前，需要明确的是，在 Python 中，一切皆对象，包括我们先前用到的字符串常量、整型常量等都是对象，变量中存放的是对象的引用。验证如下：

```
>>>print(id(5))
1482897232
>>>print(id('python'))
13845280
>>>x=2
>>>print(id(x))
1482897184
>>>y='hello'
>>>print(id(y))
57256960
```

```
>>>
```

先解释一下函数 id()的作用。id(object)返回对象 object 在其生命周期内位于内存中的地址，id 函数的参数类型是一个对象，因此，由于语句 id(5)没有报错，就可以知道 5 在这里是一个对象。又如：

```
x=2
print(id(2))
print(id(x))
y='hello'
print(id('hello'))
print(id(y))
```

其运行结果如下：

```
>>>
1683699488
1683699488
58192800
58192800
>>>
```

从结果可以看出，id(x)和 id(2)的值是一样的，id(y)和 id('hello')的值也是一样的。

在 Python 中，一切皆对象。像 2、'hello'这样的值都是对象，只不过 2 是一个整型对象，而'hello'是一个字符串对象。上面的 x=2，在 Python 中实际的处理过程是这样的：先申请一段内存分配给一个整型对象来存储整型值 2，然后让变量 x 去指向这个对象，实际上就是指向这段内存。而 id(2)和 id(x)的结果一样，说明 id 函数在作用于变量时，其返回的是变量指向的对象的地址。因为变量也是对象，所以在这里可以将 x 看成是对象 2 的一个引用。

下面再看个例子：

```
x=2
print(id(x))
y=2
print(id(y))
s='hello'
print(id(s))
t=s
print(id(t))
```

其运行结果如下：

```
>>>
```

```
1683699488
1683699488
58586016
58586016
>>>
```

从运行结果可以看到，id(x)和 id(y)的结果是相同的，id(s)和 id(t)的结果也是相同的。这说明 x 和 y 指向的是同一对象，而 t 和 s 也是指向同一对象。x=2 这句让变量 x 指向了 int 类型的对象 2，而 y=2 这句执行时，并不重新为 2 分配空间，而是让 y 直接指向了已经存在的 int 类型的对象 2。这个很好理解，因为本身只是想给 y 赋一个值 2，而在内存中已经存在了这样一个 int 类型的对象 2，所以就直接让 y 指向了已经存在的对象。这样一来，不仅能达到目的，还能节约内存空间。t=s 这句变量互相赋值，也相当于是让 t 指向了已经存在的字符串类型的对象'hello'。

下面就来讨论一下函数的参数传递和改变这个问题。

在 Python 中，参数传递采用的是值传递。先看个例子：

```
def modify1(m,K):
    m=2
    K=[4,5,6]
    return

def modify2(m,K):
    m=2
    K[0]=0
    return

n=100
L=[1,2,3]
modify1(n,L)
print(n)
print(L)
modify2(n,L)
print(n)
print(L)
```

程序运行结果如下：

```
>>>
100
[1, 2, 3]
100
[0, 2, 3]
```

```
>>>
```

从结果可以看出，执行 modify1()之后，n 和 L 都没有发生任何改变；执行 modify2()后，n 还是没有改变，但 L 发生了改变。因为在 Python 中，参数传递采用的是值传递方式，在执行函数 modify1 时，先获取 n 和 L 的 id()值，然后为形参 m 和 K 分配空间，让 m 和 K 分别指向对象 100 和对象[1,2,3]。m=2 这句让 m 重新指向对象 2，而 K=[4,5,6]这句让 K 重新指向对象[4,5,6]。这种改变并不会影响到实参 n 和 L，所以在执行 modify1 之后，n 和 L 没有发生任何改变。在执行函数 modify2 时，同理，让 m 和 K 分别指向对象 2 和对象[1,2,3]，然而 K[0]=0 让 K[0]重新指向了对象 0(注意这里 K 和 L 指向的是同一段内存)，所以对 K 指向的内存数据进行的任何改变也会影响到 L，因此在执行 modify2 后，L 发生了改变。

3.3.5 变量的作用域

在 Python 中，也存在作用域的问题，会为每个层次生成一个符号表，里层能调用外层中的变量，而外层不能调用里层中的变量，并且当外层和里层有同名变量时，外层变量会被里层变量屏蔽掉。

【例 3-7】不同作用域中的变量：

```
def function():
    x=2
    count=2
    while count>0:
        x=3
        print(x)
        count -= 1

function()
```

运行结果如下：

```
3
3
>>>
```

在函数 function 中，while 循环的外部和内部都有变量 x，此时 while 循环外部的变量 x 会被屏蔽掉。注意在函数内部定义的变量作用域都仅限于函数内部，在函数外部是不能够调用的，一般称这种变量为局部变量。

还有一种变量叫作全局变量，它是在函数外部定义的，作用域是整个程序。全局变量可以直接在函数内部应用，但是，如果要在函数内部改变全局变量，必须使用 global 关键

字进行声明。

【例 3-8】全局变量：

```
x=2
def fun1():
    print(x)

def fun2():
    global x        # global 语句用于声明一个或多个全局变量
    x=3
    print(x)

fun1()
fun2()
print(x)
```

运行结果如下：

```
2
3
3
>>>
```

函数 def 定义的变量只能在 def 的内部使用，不能在函数外部使用。一个在 def 之外被赋值的变量 X 与一个在 def 内部被赋值的变量 X 是完全不同的两个变量。Python 变量可以分为本地(def 内部，除非用 global 声明)、全局(模块内部)和内置(预定义的__builtin__模块)。全局声明 global 会将变量名映射到模块文件内部的作用域。变量名的引用将依次查找本地、全局、内置变量。例如：

```
X = 99
def add(Y):
    Z = X + Y
    return Z

print(add(1))
```

结果如下：

```
100
>>>
```

global 语句用于声明一个或多个全局变量。例如：

```
X = 88
def func():
```

```
    global X
    X = 99

func()
print(X)
```

输出结果如下：

```
99
>>>
```

再例如：

```
y,z = 1,2

def func():
    global x
    x = y + z

func()
print(x,y,z)
```

结果如下：

```
>>> 3 1 2
>>>
```

3.3.6 函数参数的类型

先前我们接触到的函数的参数叫作位置参数，即参数是通过位置进行匹配的，从左到右依次进行，对参数的位置和个数都有严格的要求。而在 Python 中，还有一种是通过参数的名称来匹配，这种匹配方式不需要严格按照参数定义时的位置来传递，这种参数叫作关键字参数。

例如：

```
def display(a,b):
    print(a)
    print(b)

display('hello ','world')
```

这段程序是想输出 hello world，可以正常运行。如果是下面这段代码，可能就得不到预期的结果：

```
def display(a,b):
    print(a)
    print(b)

display('hello')              #这样会报错
display('world','hello')      #这样会输出 worldhello
```

可以看出，在 Python 中默认是采用位置参数来匹配，所以在调用函数时必须严格按照函数定义时参数的个数和位置来传递，否则将会出现预想不到的结果。

下面这段代码采用关键字参数传递：

```
def display(a,b):
    print(a)
    print(b)

#下面两句代码的效果是相同的
display(a='world',b='hello')
display(b='hello',a='world')
```

输出结果如下：

```
world
hello
world
hello
>>>
```

从上面的输出结果可知，通过指定参数名称传递参数时，参数位置对结果没有影响。另外，关键字参数最优越的地方在于它能够给函数参数提供默认值。例如：

```
def display(a='hello',b='world'):
    print(a+b)

display()
display(b='world')
display(a='hello')
display('world')
```

输出结果如下：

```
helloworld
helloworld
helloworld
worldworld
>>>
```

在上面的代码中，分别给 a 和 b 指定了默认参数，即如果不给 a 或 b 传递参数，它们就分别采用默认值。在给参数指定了默认值后，如果传递参数时不指定参数名，则会从左到右依次传递参数，比如 display('world')没有指定'world'是传递给 a 还是 b，则默认从左向右匹配，即传递给 a。另外，默认参数一般靠右。

使用默认参数固然方便，但在重复调用函数时，默认形参会继承前一次调用结束之后该形参的值。例如：

```
def insert(a,L=[]):
    L.append(a)
    print(L)

insert('hello')
insert('world')
```

其运行结果如下：

```
['hello']
['hello', 'world']
>>>
```

3.3.7 任意个数的参数

一般情况下，在定义函数时，函数参数的个数是确定的，然而，在某些情况下，参数的个数是不确定的。比如某系统要存储用户的姓名及其小名，有些人的小名可能有两个或者更多个，此时无法确定参数的个数，就可以使用任意多个参数(收集参数)，使用收集参数时，只需在参数前面加上“*”或者“**”即可。例如：

```
def storename(name,*nickName):
    print('real name is %s' %name)
    for nickname in nickName:
        print(nickname)

storename('jack')
storename(u'詹姆斯',u'小皇帝')
storename(u'奥尼尔',u'大鲨鱼',u'三不沾')
```

输出结果如下：

```
real name is jack
real name is 詹姆斯
小皇帝
real name is 奥尼尔
```

```
大鲨鱼
三不沾
>>>
```

“*”和“**”表示能够接受 0 到任意多个参数，“*”表示将没有匹配的值都放在同一个元组中，“**”表示将没有匹配的值都放在一个 dictionary 中。例如：

```
def printvalue(a,*s,**d):
    print(a,s,d)

printvalue(1,2,c=3)
printvalue(1,3,4,2,c=3,f="l")
```

输出结果如下：

```
1 (2,) {'c': 3}
1 (3, 4, 2) {'c': 3, 'f': 'l'}
>>>
```

需要补充一点：在 Python 中，函数是可以返回多个值的，如果返回多个值，会将多个值放在一个元组或者其他类型的集合中返回。例如：

```
def function():
    x=2
    y=[3,4]
    return x,y

print(function())
```

输出结果如下：

```
(2, [3, 4])
>>>
```

3.3.8　函数调用

把已经编辑好的程序代码保存成.py 文件，就可以直接在 Python 的 IDLE 中运行了，从菜单栏中选择 File → Open…命令打开即可，再直接按 F5 键即可运行。

Python 可以调用已保存为 a.py 文件内的所有函数，方案如下。

(1)　将 a.py 文件做成一个包，或者直接和调用文件放在同一个目录下。

(2)　在调用文件头引入：from a import *。

这样就可以使用 a.py 文件内的所有函数了。

【例 3-9】在 prin.py 文件中调用 tel.py 文件。

文件 tel.py 的代码内容：

```
name=["Ben","Jone","John","Jerry","Anny","Ivy","Jan","Wong"]
tel=[6601,6602,6603,6604,6605,6606,6607,6608]

Tellbook={}
for i in range(len(name)):
    d1="{}".format(name[i])
    d2="{}".format(tel[i])
    Tellbook[d1]=d2
```

文件 prin.py 的代码内容：

```
from tel import *
print(Tellbook)
print('this is a test for import.')
```

prin.py 文件要做两件事情，先把 tel.py 文件中的 Tellbook 打印一下，再打印一句话：‘this is a test for import.’。

执行文件 prin.py，其结果如下：

```
{'John': '6603', 'Ben': '6601', 'Jone': '6602', 'Ivy': '6606', 'Jerry':
'6604', 'Wong': '6608', 'Jan': '6607', 'Anny': '6605'}
this is a test for import.
>>>
```

函数也一样，当我们调用函数时，也需要使用 **import** 来导入。

【例 3-10】函数的导入和调用。addyu.py 文件的内容如下：

```
#addyu.py
def add(a=0,b=0):
    '''
    此函数是计算两个数的和
    当不输入参数时,默认的是 0+0
    '''
    c=a+b
    print(c)
def test(m,K=0,*tup,**dic):
    print('m:',m)
    print('K:',K)
    print('tup:',tup)
    print('dic:',dic)
    return
```

在下面的 test_addyu.py 文件中调用 addyu.py 内的 add(a,b)函数：

```
#test_addyu.py
from addyu import add
a=add(1,2)
```

这里的 from addyu import add 的意思是从 addyu.py 文件中导入 add(a,b)函数。当然，如果要导入 addyu.py 文件中所有的函数，那就是 from addyu import *，为了避免导入所有的函数占内存，所以一般用到什么函数就导入什么函数。只有当导入的函数比较多时，才使用*。另外，为了防止导入的多个模块中有相同的函数名而引起混乱，不建议使用*。

执行 test_addyu.py，结果如下：

```
3
>>>
```

说明： 写函数时要养成好的习惯——写函数文档，即写清楚函数的功能是什么、怎么用，并把文档内容用三引号注释起来，它不是函数体的执行代码，它的作用是给 help 提供查询的，如查上例中 addyu.py 文件的功能：

```
>>> help(add)
Help on function addyu in module addyu:

add(a=0, b=0)
    此函数是计算两个数的和
    当不输入参数时,默认的是 0+0
>>>
```

还是上例，按如下输入，并观察结果：

```
>>> from addyu import test
>>> test('a1')
m: a1
K: 0
tup: ()
dic: {}
>>>

>>> test(1,3)
m: 1
K: 3
tup: ()
dic: {}
>>>
```

```
>>> test('a1',K=2)
m: a1
K: 2
tup: ()
dic: {}
>>>

>>> test('a1',2,3,6,fname='yu',name='bg')
m: a1
K: 2
tup: (3, 6)
dic: {'fname': 'yu', 'name': 'bg'}
>>>

>>> test('a1',K=2,3,6,fname='yu',name='bg')
SyntaxError: positional argument follows keyword argument
>>> test('a1',K=2,fname='yu',name='bg')
m: a1
K: 2
tup: ()
dic: {'fname': 'yu', 'name': 'bg'}
>>> test(K=2,fname='yu',name='bg')
Traceback (most recent call last):
  File "<pyshell#14>", line 1, in <module>
    test(K=2,fname='yu',name='bg')
TypeError: test() missing 1 required positional argument: 'm'
>>>
```

请比较一下出错的原因。

关于函数名的一个补充：函数名其实就是指向一个函数对象的引用，完全可以把函数名赋给一个变量，相当于给这个函数起了一个“别名”。例如：

```
>>> a = int     #变量 a 指向 int 函数
>>> a('2')      #可以通过 a 调用 int 函数
2
>>>
```

3.4　函数式编程

3.4.1　lambda

lambda 是匿名函数，也叫行内函数。

具体用法如下：

```
f = lambda x : x+2    #定义了一个函数 f(x)=x+2
g = lambda x,y : x+y  #定义了一个函数 f(x,y)=x+y
```

lambda 只是一个表达式，函数体比 def 简单很多。lambda 表达式运行起来像一个函数。关于 lambda 函数的用途有两点说明。

(1) 对于单行函数，使用 lambda 可以省去定义函数的过程，让代码更加精简。

(2) 在非多次调用函数的情况下，lambda 表达式即用即得，可以提高性能。

注意： 如果是 for...in...if(后面将会讲到)能做的，最好不要选择 lambda。

使用 lambda 匿名函数的例子如下：

```
>>> f = lambda x,y,z:x+y+z
>>> f(1,2,3)
6
>>>
```

3.4.2 reduce()

使用 reduce(func(), Z)，可以把一个函数作用在一个序列上累计计算。其效果如下：

```
reduce(f, [x1, x2, x3, x4]) = f(f(f(x1, x2), x3), x4)
```

Python 中的 reduce 内建函数 func()是一个二元操作函数，它将一个数据集合 Z(可以是列表或元组等)中的所有数据进行下列操作：用函数 func()对集合 Z 中的第 1、第 2 个数据进行运算，将得到的结果再与 Z 中的第三个数据运算，以此循环。

例如：

```
from functools import reduce
def add(x,y):
   return x+y

sum=reduce(add,(1,2,3,4,5,6,7))
print(sum)
```

执行上面的程序，输出 1+2+3+4+5+6+7 的结果即 28。reduce 就是将 add(x,y)函数作用在(1,2,3,4,5,6,7)上，即：1 赋给 x，2 赋给 y，再将计算结果(1+2)赋给 x，3 赋给 y；再次将计算结果((1+2)+3)赋给 x，4 赋给 y，依此类推。

当然，也可以用 lambda 的方法，更为简单：

```
>>> from functools import reduce
>>> sum=reduce(lambda x,y:x+y,(1,2,3,4,5,6,7))
```

```
>>> print(sum)
28
>>>
```

reduce 在 Python 2.x 中可以直接用，但在 Python 3.0 以后，已经不在 built-in function 中，要使用它就必须先导入：from functools import reduce。

3.4.3 filter()

filter()用于过滤序列。filter()接收一个函数和一个序列，并把传入的函数依次作用于每个元素，然后根据返回值 True 或者 False，决定保留还是丢弃该元素。

例如，在一个 list 中，删掉偶数，只保留奇数，代码如下：

```
>>> def is_odd(n):
        return n % 2 == 1
>>> list(filter(is_odd, [1, 2, 4, 5, 6, 9, 10, 15]))
[1, 5, 9, 15]
>>>
```

再如把一个序列中的空字符串删掉，代码如下：

```
>>> def not_empty(s):
        return s and s.strip()
>>> list(filter(not_empty, ['A', '', 'B', None, 'C', '  ']))
['A', 'B', 'C']
>>>
```

可见，用 filter()这个高阶函数时，关键在于正确地实现一个“筛选”函数。

filter()函数还可以实现行函数功能。例如：

```
>>> b=[i for i in range(1,10) if i>5 and i<8]
>>> b
[6, 7]
>>>
```

用 filter()函数改写如下：

```
>>> b=filter(lambda x: x>5 and x<8, range(1,10))
>>> list(b)
[6, 7]
>>>
```

filter()在 Python 2.7 中直接返回一个列表。但在 Python 3.x 中，filter()是一个容器，返回时需要用 list 调用才显示数据。

3.4.4　map()

map(func, S)将传入的函数 func 依次作用到序列 S 的每个元素，并把结果作为新的序列返回。

函数 func 在 S 域上遍历，在 Python 3.x 中 map()是一个容器，返回时需要用 list 调用才显示数据，显示的是 func 作用后的结果数据。

【例 3-11】比较 map 和 filter：

```
>>> list(map(lambda x:x**2,[1,2,3]))
[1, 4, 9]
>>> list(filter(lambda x:x**2,[1,2,3]))
[1, 2, 3]
>>>
```

说明： map 返回的是 func 作用后的结果数据，而 filter 是通过 func 作用筛选出作用域的数据。map 还可以接受多个参数的函数，例如：

```
>>> list(map(lambda x,y:x*y+x,[1,2,3],[4,5,6]))
[5, 12, 21]
>>>
```

map()、filter()、reduce()三个函数循环要比 for、while 循环快得多。

3.4.5　行函数

行函数也叫列表解析或列表推导式，格式如下：

```
[<expr1> for k in L if <expr2>]
```

【例 3-12】将列表中能被 2 整除的元素提取出来并加上 2：

```
list=[1,2,3]
A=[k+2 for k in list if k % 2 == 0]
print(A)
```

可用一行代码实现：

```
>>> [k+2 for k in [1,2,3] if k % 2 == 0]
[4]
>>>
```

列表推导式(list comprehension)是利用其他列表创建新列表(类似于数学术语中的集合推导式)的一种方法。它的工作方式类似于 for 循环。例如：

```
>>> [x*x for x in range(10)]
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
>>>
```

如果只想打印出那些能被 3 整除的平方数，只需要通过添加一个 if 部分在推导式中就可以完成：

```
>>> [x*x for x in range(10) if x % 3 == 0]
[0, 9, 36, 81]
>>>
```

也可以增加更多的 for 语句部分。例如：

```
>>> [(x,y) for x in range(3) for y in range(3)]
[(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2)]
>>> [[x,y] for x in range(2) for y in range(2)]
[[0, 0], [0, 1], [1, 0], [1, 1]]
>>>
```

3.5　常用的内置函数

前面已经介绍过 dir、help、len、id、type、range 等内置函数，除此而外，常见的内置函数还有 sum、zip、enumerate、max、min 等。

3.5.1　sum

sum(s)是 Python 中一个很实用的函数，但要注意参数 s 的类型。

例如：

```
>>> s = sum(1,2,3)
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    s = sum(1,2,3)
TypeError: sum expected at most 2 arguments, got 3
>>>
```

其实 sum()的参数是一个 list。例如：

```
>>> sum([1,2,3])
6
>>> sum(range(1,11))
55
>>>
```

sum 还有一个用法：

```
>>> a = range(1,11)
>>> b = range(1,10)
>>> c =  sum([item for item in a if item in b])
>>> print(c)
45
>>>
```

其实，sum 的参数也可以是集合、元组，甚至可以是字典。当然，其计算的对象都应该是数值型 int。例如：

```
>>> a={1,2,3}
>>> sum(a)
6
>>> b=(1,2,3)
>>> sum(b)
6
>>> c={1:3,2:4,5:6}
>>> sum(c)          对字典计算的是键 key，不是 value
8
>>> sum(c.values())
13
>>>
```

从上面可以看出，对字典用 sum 计算的是 key，而不是 value。计算 value 值时，使用字典对象的属性 values()。

3.5.2　zip

zip(t,s)返回 t 和 s 的一个相互匹配的列表。例如：

```
>>> t='abc'
>>> s=[1,2,3]
>>> z=zip(t,s)
<zip object at 0x04104A08>
>>> list(z)
[('a', 1), ('b', 2), ('c', 3)]
>>>
```

若长度不够，以最短的为主。例如：

```
>>> zip('abcd','123')
<zip object at 0x041049E0>
>>> list(zip('abcd','123'))
```

```
[('a', '1'), ('b', '2'), ('c', '3')]
>>>
```

这类似于元组里说的解包。

再如：

```
>>> x = [1, 2, 3]
>>> y = [4, 5, 6]
>>> z = [7, 8, 9]
>>> xyz = zip(x, y, z)
>>> u = zip(*xyz)
>>> print(list(u))
[(1, 2, 3), (4, 5, 6), (7, 8, 9)]
>>>
```

一般认为*是一个 unzip 的过程，它的运行机制如下。

在运行 zip(*xyz)之前，xyz 的值是[(1, 4, 7), (2, 5, 8), (3, 6, 9)]，那么 zip(*xyz)等价于 zip((1, 4, 7), (2, 5, 8), (3, 6, 9))，所以运行结果是[(1, 2, 3), (4, 5, 6), (7, 8, 9)]。

试试下面的练习：

```
>>> x = [1, 2, 3]
>>> r = zip(* [x] * 3)
>>> print(list(r))
[(1, 1, 1), (2, 2, 2), (3, 3, 3)]
>>>
```

上面的代码运行机制是这样的：

- [x]生成一个元素的列表，它只有一个元素 x——列表。
- [x] * 3 生成一个列表，它有 3 个元素[x, x, x]。
- zip(* [x] * 3)的意思就明确了，zip(x, x, x)。

dict 和 zip 可以组合使用，生成字典。例如：

```
>>> d= dict(zip('abc', range(1,4)))
>>> d
{'b': 2, 'c': 3, 'a': 1}
>>>
```

3.5.3 enumerate

enumerate(t)返回 t 的 index 和元素对，t 可以是字符串、列表、元组、字典等，若是字典，则返回的是键名。例如：

```
>>> t={'first':'j','second':'h','third':'abc'}
>>> for i,k in enumerate(t):
print(i,k)

0 first
1 third
2 second
>>>
```

3.5.4　max 和 min

max()和 min()的参数同 sum()，可以是列表、元组、集合、字典，甚至可以是 range()。

当然，字典默认的是针对 key，若需要对 value 进行计算，则需要使用.values()方法。

例如：

```
>>> a={1:2,3:4,5:6}
>>> max(a)
5
>>> min(a.values())
2

>>> B={'a':2,'b':4,'c':6}
>>> min(B)
 'a'
>>>
```

当 key 不是数值型时，比较的是 key 的 ASCII 码。

3.5.5　eval

eval()将字符串 str 当成有效的表达式来求值，并返回计算结果。例如：

```
>>>x = 1
>>>eval('x+1')
2
>>>a = "[[1,2], [3,4], [5,6], [7,8], [9,0]]" #注意 a 是字符串
>>>b = eval(a)
>>>b
[[1, 2], [3, 4], [5, 6], [7, 8], [9, 0]]
>>>type(b)
list
```

```
>>>c = "{1: 'a', 2: 'b'}" #注意 c 是字符串
>>>d = eval(c)
>>>d
 {1: 'a', 2: 'b'}
>>>type(d)
dict
>>>e = "([1,2], [3,4], [5,6], [7,8], (9,0))"
>>>f = eval(e)
>>>f
 ([1, 2], [3, 4], [5, 6], [7, 8], (9, 0))
>>>
```

但下面的代码存在很大的风险：

```
>>> __import__('os').system('dir >dir.txt')
0
>>> open('dir.txt').read()
' 驱动器 D 中的卷是 soft\n 卷的序列号是 0046-527B\n\n D:\\python35 的目录
\n\n2016-08-13  01:19    <DIR>          .\n2016-08-13  01:19
<DIR>          ..\n2016-06-21  22:51               406 0.py\n2016-08-13
01:19                0 dir.txt\n2016-06-15  22:07    <DIR>
DLLs\n2016-06-15  22:07    <DIR>          Doc\n2016-06-15  22:06    <DIR>
include\n2016-06-15  22:07    <DIR>          Lib\n2016-06-15  22:07    <DIR>
libs\n2015-12-07  19:56           30,338 LICENSE.txt\n2015-12-06  06:32
310,468 NEWS.txt\n2015-12-06  01:40           38,680 python.exe\n2015-12-
06  01:39           51,480 python3.dll\n2015-12-06  01:39        3,122,968
python35.dll\n2015-12-06  01:40           38,680 pythonw.exe\n2015-11-22
22:58            8,269 README.txt\n2016-06-15  22:08    <DIR>
Scripts\n2016-06-15  22:08    <DIR>          tcl\n2016-06-25  11:37
317 test.py\n2016-06-25  11:37              317 test1.py\n2016-06-25
18:59              150 TestClass.py\n2016-06-15  22:07    <DIR>
Tools\n2015-06-25  23:34           85,328 vcruntime140.dll\n2016-06-25
11:38    <DIR>          __pycache__\n              13 个文件      3,687,401 字节
\n             11 个目录 35,201,863,680 可用字节\n'
>>>
```

执行上面的两句代码，其实就已经在 Python 3.x 安装目录下建立了一个名为 dir.txt 的文件。如若再运行下面这两句代码，则可以将新建的 dir.txt 文件删除！

```
>>> import os  #导入 os 模块
>>> os.system('del dir.txt /q')
0
>>>
```

上面新建的 dir.txt 文件已经被删除了！也就是说，这两句代码可以删除本台计算机上

的任何文件！

下面的代码请自行测试：

```
>>>eval("__import__('os').system(r'md c:\\testtest')")
>>>eval("__import__('os').system(r'rd/s/q c:\\testtest')")
>>>eval("__import__('os').startfile(r'c:\windows\\notepad.exe')")
>>>
```

3.5.6　判断函数

下面这几个函数常用于判断。

(1)　**in**：常用于字符串、列表、元组、字典、集合中，用来判断一个字符串或者一个元素是否属于字符串或者列表等。同样，对应的还有 not in。

例如：

```
>>>a={'a':2,'b':4,'c':6}
>>>'a' in a
True
>>>'c' not in a
False
>>>
```

(2)　**startswith()**、**endswith()**：做文本处理时，用来判断字符串开始和结束的位置。基本格式如下：

```
S.startswith(prefix[, start[, end]])
S.endswith(suffix[, start[, end]])
```

示例代码如下：

```
>>>"fish".startswith('fi')
True
>>>"fish".startswith('fi',1)  #此处的 1 表示从索引位置 1 开始
False
>>> "fish".endswith('sh')
True
>>> "fish".endswith('sh',3)
False
>>>
```

Python 的这两个函数有个特别的地方——它的 prefix 和 suffix 参数不仅可以是字符串，还都可以是一个元组。只要其中一个成立，就返回 True，也就是一种“或”的关系。

例如：

```
if filename.endswith(('.gif', '.jpg', '.tiff')):
    print("%s 是一个图片文件" %filename)
```

上面两行代码根据文件扩展名是否是 gif、jpg 或 tiff 之一来确定文件是不是图片文件。这个代码也可以写成：

```
if filename.endswith(".gif") or filename.endswith(".jpg") or \ #续行
filename.endswith(".tiff"):
    print("%s 是一个图片文件"%filename)
```

不过，这样比较麻烦。值得注意的是，别忘了元组的括号。

(3) **isalnum()**：检测字符串是否仅由字母和数字组成，若其间夹杂有空格、标点符号或者其他字符，则都返回 False。例如：

```
>>>str = "this2009"
>>>print(str.isalnum())
True
>>>str = "this is string example....wow!!!"
>>>print(str.isalnum())
False
>>>str1 = "hello"
>>>print(str1.isalnum())
True
>>>
```

(4) **isaipha()**：检测字符串是否只由字母组成。如果字符串中的所有字符都是字母，则返回 True，否则返回 False。例如：

```
>>>str = "this"
>>>print(str.isalpha())
True
>>>str = "this is string example....wow!!!"
>>>print(str.isalpha())
False
>>>
```

(5) **isdigit()**：检查字符串是否只包含数字(全由数字组成)。如果字符串中的所有字符都是数字，则返回 True，否则返回 False。例如：

```
>>>str = "123456"
>>>print(str.isdigit())
True
>>>str = "this is string example....wow!!!"
```

```
>>>print(str.isdigit())
False
>>>
```

其他内置函数见表 3-2。

表 3-2　其他内置函数

内置函数	说　明
abs(x)	求绝对值 ① 参数可以是整型，也可以是复数 ② 若参数是复数，则返回复数的模
divmod(a, b)	分别取 a/b 的商和余数 注意：整型、浮点型都可以
int([x[, base]])	将一个字符转换为 int 类型，base 表示进制
pow(x, y[, z])	返回 x 的 y 次幂
range([start], stop[, step])	产生一个序列，默认从 0 开始
round(x[, n])	四舍五入
sum(iterable[, start])	对集合求和
chr(i)	返回整数 i 对应的 ASCII 字符
bin(x)	将整数 x 转换为二进制字符串
format(value [, format_spec])	格式化输出字符串 格式化的参数顺序从 0 开始，如“I am {0},I like {1}”
enumerate(sequence [, start = 0])	返回一个可枚举的对象，其 next()方法将返回一个 tuple
max(iterable[, args...][key])	返回集合中的最大值
min(iterable[, args...][key])	返回集合中的最小值
dict([arg])	创建数据字典
list([iterable])	将一个集合类转换为另外一个集合类
set()	set 对象实例化
str([object])	转换为 string 类型
sorted(iterable[, cmp[, key[, reverse]]])	对集合列表排序，不对原集合列表排序，返回新的集合列表
tuple([iterable])	生成一个 tuple 类型
dir([object])	① 不带参数时，返回当前范围内的变量、方法和定义的类型列表 ② 带参数时，返回参数的属性、方法列表 ③ 当参数为实例时，如果参数包含方法__dir__()，该方法将被调用 ④ 如果参数不包含__dir__()，该方法将最大限度地收集参数信息

续表

内置函数	说　明
eval(expression [, globals [, locals]])	计算表达式 expression 的值
id(object)	返回对象的唯一标识
len(s)	返回集合长度
locals()	返回当前的变量列表
map(function, iterable, ...)	遍历每个元素，执行 function 操作
next(iterator[, default])	类似于 iterator.next()
reduce(function, iterable[, initializer])	合并操作，从第一个开始是前两个参数，然后是前两个的结果与第三个合并进行处理，以此类推
type(object)	返回该 object 的类型
zip([iterable, ...])	矩阵变幻处理
file(filename [, mode [, bufsize]])	file 类型的构造函数，作用为打开一个文件，如果文件不存在且 mode 为写或追加时，文件将被创建。添加‘b’到 mode 参数中，将对文件以二进制形式操作。添加‘+’到 mode 参数中，将允许对文件同时进行读写操作
	① 参数 filename：文件名称
	② 参数 mode：‘r’(读)、‘w’(写)、‘a’(追加)
	③ 参数 bufsize：如果为 0，表示不进行缓冲；如果为 1，表示进行缓冲，如果是一个大于 1 的数，表示缓冲区的大小
input([prompt])	获取用户输入，返回的结果是字符型
open(name[, mode[, buffering]])	打开文件
	思考与 file 有什么不同？推荐使用 open
print()	打印函数

3.6　常见的错误显示

Python 程序常见的异常见表 3-3。

表 3-3　常见的错误类型

异　常	描　述
NameError	尝试访问一个没有声明的变量
SyntaxError	语法错误
ZeroDivisionError	除数为 0
IndexError	索引超出序列范围
KeyError	请求一个不存在的字典关键字

续表

异　常	描　述
IOError	输入输出错误(比如要读的文件不存在)
AttributeError	尝试访问未知的对象属性
IndentationError	冒号换行后没有缩进
ValueError	传给函数的参数类型不正确，比如给 int()函数传入字符串型

一个常见错误的例子：

```
while 1:
    try:
    x = int(input('No.1: '))
    y = int(input('No.2: '))

    r= x/y
    print(r)
    except (Exception) as e:
        print(e)
        print('again')
    else:
        break

  File "<ipython-input-15-6f26b9b9c36e>", line 3
    x = int(input('No.1: '))
    ^
IndentationError: expected an indented block
```

上面的错误是因为 try 行后有冒号，下一行代码应该有缩进，包括后面的几行都应该缩进。

3.6.1　常见的错误类型

(1)　NameError：尝试访问一个未声明的变量。例如：

```
>>> v
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    v
NameError: name 'v' is not defined
>>>
```

(2)　ZeroDivisionError：除数为 0。例如：

```
>>> v = 1/0
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    v = 1/0
ZeroDivisionError: division by zero
>>>
```

(3) SyntaxError：语法错误。例如：

```
>>> int 4
SyntaxError: invalid syntax
>>>
```

(4) IndexError：索引超出范围。例如：

```
>>> list1 = ['s','d']
>>> list1[3]
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    list1[3]
IndexError: list index out of range
>>>
```

(5) KeyError：字典关键字不存在。例如：

```
>>> Dic = {'1':'yes', '2':'no'}
>>> Dic['3']
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    Dic['3']
KeyError: '3'
>>>
```

(6) AttributeError：访问未知对象属性。例如：

```
>>> str='123'
>>> str.append()
Traceback (most recent call last):
  File "<pyshell#9>", line 1, in <module>
    str.append()
AttributeError: 'str' object has no attribute 'append'
>>>
```

(7) ValueError：数值错误。例如：

```
>>> int('d')
Traceback (most recent call last):
  File "<pyshell#10>", line 1, in <module>
```

```
    int('d')
ValueError: invalid literal for int() with base 10: 'd'
>>>
```

(8)　TypeError：类型错误。例如：

```
>>> print('123'+45)
Traceback (most recent call last):
  File "<pyshell#11>", line 1, in <module>
    print('123'+45)
TypeError: Can't convert 'int' object to str implicitly
>>>
```

3.6.2　初学者常犯的错误

初学 Python 时，想要彻底弄懂 Python 错误信息的含义可能有点不容易，下面列出了程序运行时出现的一些常见异常，这些异常也是新手常犯的错误。

(1)　忘记在 if、elif、else、for、while、class、def 等末尾添加冒号(:)，导致出现了“SyntaxError: invalid syntax”错误。

该错误将发生在类似如下的代码中：

```
if spam == 42
    print('Hello!')
```

(2)　使用 = 而不是 ==，导致出现了“SyntaxError: invalid syntax”错误。

= 是赋值操作符而 == 是等于比较操作。该错误发生在类似如下的代码中：

```
if spam = 42:
    print('Hello!')
```

(3)　错误地使用了缩进量，导致出现了“IndentationError: unexpected indent”、“IndentationError: unindent does not match any outer indetation level”以及“IndentationError: expected an indented block”错误。记住缩进只用在以“:”结束的语句之后，而之后必须恢复到之前的缩进格式，缩进是四个空格。该错误发生在类似如下的代码中：

```
print('Hello!')
   print('Howdy!')
```

或者：

```
if spam == 42:
    print('Hello!')
 print('Howdy!')
```

或者：

```
if spam == 42:
print('Hello!')
```

缩进一定要按层次缩进，尽管 Python 3.5 已经做了很多优化，如“:”后换行有时候可以只空一个空格，也能执行，但为了保持代码的可读性，还是应尽可能地空四格，以养成良好的编写代码习惯。

(4) 在 for 循环语句中忘记调用 len()，导致出现了“TypeError: ‘list’ object cannot be interpreted as an integer”错误。

通常想要通过索引来迭代一个 list 或者 string 的元素时，需要调用 range()函数。要返回 len 值而不是返回这个列表。该错误发生在类似如下的代码中：

```
spam = ['cat', 'dog', 'mouse']
for i in range(spam):
print(spam[i])
```

(5) 尝试修改 string 的值，导致出现了“TypeError: ‘str’ object does not support item assignment”错误。

string 是一种不可变的数据类型，该错误发生在类似如下的代码中：

```
spam = 'I have a pet cat.'
spam[13] = 'r'
```

而实际想要这样做：

```
spam = 'I have a pet cat.'
spam = spam[:13] + 'r' + spam[14:]
```

(6) 尝试连接非字符串值与字符串，导致出现了“TypeError: Can’t convert ‘int’ object to str implicitly”错误。

该错误发生在类似如下的代码中：

```
numEggs = 12
print('I have ' + numEggs + ' eggs.')
```

而实际想要这样做：

```
numEggs = 12
print('I have ' + str(numEggs) + ' eggs.')
```

或者：

```
numEggs = 12
print('I have %s eggs.' % (numEggs))
```

(7) 在字符串首尾忘记加引号，导致出现了“SyntaxError: EOL while scanning string literal”错误。

该错误发生在类似如下的代码中：

```
print(Hello!')
```

或者：

```
print('Hello!)
```

或者：

```
myName = 'Al'
print('My name is ' + myName + . How are you?')
```

(8) 变量或者函数名拼写错误，导致出现了“NameError: name ‘fooba’ is not defined”错误。

该错误发生在类似如下的代码中：

```
foobar = 'Al'
print('My name is ' + fooba)
```

或者：

```
spam = ruond(4.2)
```

或者：

```
spam = Round(4.2)
```

(9) 方法名拼写错误，导致出现了“AttributeError: ‘str’ object has no attribute ‘lowerr’”错误。该错误发生在类似如下的代码中：

```
spam = 'THIS IS IN LOWERCASE.'
spam = spam.lowerr()
```

(10) 引用超过了 list 的最大索引，导致出现了“IndexError: list index out of range”错误。该错误发生在类似如下的代码中：

```
spam = ['cat', 'dog', 'mouse']
print(spam[6])
```

(11) 使用不存在的字典键值，导致出现了“KeyError: ‘spam’”错误。

该错误发生在类似如下的代码中：

```
spam = {'cat': 'Zophie', 'dog': 'Basil', 'mouse': 'Whiskers'}
print('The name of my pet zebra is ' + spam['zebra'])
```

(12) 尝试使用 Python 关键字作为变量名，导致出现了“SyntaxError: invalid syntax”错误。Python 关键字不能用作变量名，该错误发生在类似如下的代码中：

```
class = 'algebra'
```

Python 3.x 的关键字有 and、as、assert、break、class、continue、def、del、elif、else、except、False、finally、for、from、global、if、import、in、is、lambda、None、nonlocal、not、or、pass、raise、return、True、try、while、with、yield。

(13) 尝试使用 range()创建整数列表，导致出现了“TypeError: ‘range’ object does not support item assignment”错误。

有时想要得到一个有序的整数列表，所以 range()看上去是生成此列表的不错方式。然而，range()返回的是 range 对象，而不是实际的 list 值。

该错误发生在类似如下的代码中：

```
spam = range(10)
spam[4] = -1
```

而实际想要这样做：

```
spam = list(range(10))
spam[4] = -1
```

应注意：在 Python 2.7 中，spam = range(10)是可行的，因为在 Python 2.7 中 range()返回的是 list 值，但是在 Python 3.x 中会产生以上错误。

(14) 错误地使用了++(自增)或--(自减)运算符，导致出现了“SyntaxError: invalid syntax”错误。

如果习惯于使用 C++、Java、PHP 等其他的语言，也许会想要尝试使用++(自增)或--(自减)一个变量。而在 Python 中，却并没有这样的操作符。

该错误发生在类似如下的代码中：

```
spam = 1
spam++
```

而真正想要做的是：

```
spam = 1
spam += 1
```

(15) 类中忘记为方法的第一个参数添加 self 参数，导致出现了“TypeError: myMethod() takes no arguments (1 given)”错误。

该错误发生在类似如下的代码中：

```
class Foo():
    def myMethod():
        print('Hello!')

a = Foo()
myMethod()
```

而实际上应该这样做：

```
class Foo():
    def myMethod(self):
        print('Hello!')

a = Foo()
a.myMethod()
```

3.6.3　try

在 Python 中，try-except 语句主要用于处理程序正常执行过程中出现的一些异常情况，如语法错误、数据除零错误、从未定义的变量上取值等；而 try-finally 语句则主要用于监控错误的环节。

尽管 try-except 和 try-finally 的作用不同，但是在编程实践中通常可以把它们组合在一起使用，以 try-except-else-finally 的形式来实现稳定性和灵活性更好的设计。

Python 中，try-except-else-finally 语句的完整格式如下：

```
try:
    Normal execution block
except A:
    Exception A handle
except B:
    Exception B handle
except:
    Other exception handle
else: #可无，若有必有 except x 或 except 存在，仅在 try 后无异常时执行
    if no exception, get here
finally:   #此语句务必放在最后，并且也是必须执行的语句
    print("finally")
```

说明：正常执行的程序在 try 下的 Normal execution block 块中执行，在执行过程中，如果发生了异常，则中断当前在 Normal execution block 中的执行，跳转到对应的异常处理块 except x(A 或 B)中开始执行，Python 从第一个 except x 处开始查找，如果找到了对应的

exception 类型，则进入其提供的 exception handle 中进行处理，如果没有，则依次进入第二个，如果都没有找到，则直接进入 except 块进行处理。except 块是可选项，如果没有提供，该 exception 将会被提交给 Python 进行默认处理，处理方式则是终止应用程序并打印提示信息。

如果在 Normal execution block 执行块的执行过程中没有发生任何异常，则在执行完 Normal execution block 后，会进入 else 执行块中(若存在)执行。

无论发生异常与否，若有 finally 语句，以上 try-except-else 代码块执行的最后一步总是执行 finally 所对应的代码块。

需要注意如下几个问题。

(1) 在上面所示的完整语句中，try-except-else-finally 所出现的顺序必须是 try→except x→except→else→finally，即所有的 except 必须在 else 和 finally 之前，else(若有)必须在 finally 之前，而 except x 必须在 except 之前。否则会出现语法错误。

(2) 对于上面所展示的 try-except 完整格式而言，else 和 finally 都是可选的，而不是必需的。但若存在 else，则必须放在 finally 之前，finally(如果存在)必须放在整个语句的最后位置。

(3) 在上面的完整语句中，else 语句的存在必须以 except x 或者 except 语句为前提，如果在没有 except 语句的 try block 中使用 else 语句，会引发语法错误。即 else 不能仅与 try-finally 配合使用。

【例 3-13】一个 try-except-else 的例子：

```
while 1:
   try:
      x = int(input('No.1: '))
      y = int(input('No.2: '))

      r= x/y
      print(r)
   except (Exception) as e:  #不管什么异常，都捕获给 e
      print(e)
      print('again')
   else:
      break
```

这段代码的意思是：从键盘接收两个输入(数字或者字符)，转化为数字型做除法，如果符合除法规则，就输出结果，并跳转执行 else 语句，如果不符合，有错误，则进入 except 语句，打印错误原因，并重新来一次。

看下面的输入和输出结果：

```
No.1: 1
No.2: 0
division by zero
again

No.1: 1
No.2: a
invalid literal for int() with base 10: 'a'
again

No.1: 1
No.2: 2
0.5
```

final 放在 try 的最后，不管前面发生了什么，final 后面的语句均执行。

3.6.4　assert

assert 是断言的关键字。执行该语句时，先判断表达式 expression，如果表达式为真，则什么都不做；如果表达式为假，则抛出异常。例如：

```
>>> assert len('love') == len('like')
>>> assert len('love u') == len('like')
Traceback (most recent call last):
  File "<pyshell#15>", line 1, in <module>
    assert len('love u') == len('like')
AssertionError
>>>
```

可以看出，如果 assert 后面的表达式为真，则什么都不做，如果为假，就会抛出 AssertionError 异常。

3.6.5　raise

当程序出现错误时，Python 会自动引发异常，也可以通过 raise 显式地引发异常。一旦执行了 raise 语句，raise 后面的语句将不能执行。

下面演示 raise 的用法：

```
try:
    s = None
    if s is None:
```

```
        print("s 是空对象")
        raise NameError  #如果有 NameError 异常，后面代码都将不执行
    print(len(s))
except TypeError:
    print("空对象没有长度")
```

执行结果：

```
s 是空对象
Traceback (most recent call last):

  File "<ipython-input-18-c87e44f18de5>", line 5, in <module>
    raise NameError  #如果有 NameError 异常，后面代码都将不执行

NameError
```

3.7　模块和包

先来看一个例子：

```
>>> a=[1.23e+18, 1, -1.23e+18]
>>> sum(a)
0.0
>>>
```

怎么会是 0？再执行下面的代码：

```
>>> import math
>>> math.fsum(a)
1.0
>>>
```

这就对了！

计算机由于浮点数的运算问题，会导致上面的结果有差异。但是我们引入一个 math 后，计算结果就正常了。

3.7.1　模块(module)

模块是包含函数和其他语句的 Python 脚本文件，它以“.py”为后缀名，即用 Python 脚本的后缀名。表现形式为：编写的代码保存为文件，这个文件就是一个模块，如 sample.py，其中，文件名 sample 为模块名称。

在 Python 中可以通过导入模块，然后使用其模块中提供的函数或者数据。关于模块的

导入方法，这里以 math 模块为例：

- import math：导入 math 模块。
- import math as m：导入 math 模块并取个别名为 m。
- from math import exp as e：导入 math 库中的 exp 函数并取别名为 e。

要想使用 import 导入的模块中的函数，则必须以“模块名.函数名”的形式调用函数；而 from 是将模块中某个函数而不是整个模块导入，所以使用 from 导入的模块中的某个函数，可以直接以函数名调用，不必在前面加上模块名称。例如：

```
>>> import math as m         #给 math 模块取个别名 m，使用时用 m 替代 math
>>> a=[1.23e+18, 1, -1.23e+18]
>>> m.fsum(a)
1.0

>>> from math import fsum    #这里仅导入了 math 模块中的 fsum 函数
>>> a=[1.23e+18, 1, -1.23e+18]
>>> fsum(a)                  #直接使用 fsum()函数，不再使用 math.fsum()
1.0
>>>
```

以 from 导入模块中的函数后，使用模块中的函数会方便很多，不再使用模块名。如果要想将多个模块中的所有函数都采用这种方式导入，则可以在 from 中使用“*”通配符，表示导入模块中的所有函数，但一般不这么用。如下所示：

```
>>> from math import sqrt  #仅导入了 sqrt 函数
>>> sqrt(4)
2.0
>>> cos(4)            #仅导入了 sqrt 函数，所以 cos 函数不能用
Traceback (most recent call last):
  File "<pyshell#13>", line 1, in <module>
    cos(4)
NameError: name 'cos' is not defined
>>> from math import *   #将 math 中的所有函数全部导入
>>> sqrt(4)
2.0
>>> cos(4)
-0.6536436208636119
>>>
```

为了确保在 Python 2.1 之前版本的 Python 中可以正常运行一些新的语言特性，需要导入一个包：from __future__ import *。模块就是一个扩展名为.py 的程序文件。我们可以直接引用它，节省时间和精力，无须重复写同样的代码。这也表明 Python 是开源的，符合

“他山之石可以攻玉”的思想。

我们试着写一个模块，保存为.py 文件，并调用！

新建一个 addyu.py 文件，内容如下：

```
#addyu.py
def add(a=0,b=0):
    '''
    此函数是计算两个数的和
    当不输入参数时，默认的是 0+0
    '''
    c=a+b
    print(c)
def test(m,K=0,*tup,**dic):
    print('m:',m)
    print('K:',K)
    print('tup:',tup)
    print('dic:',dic)
    return
```

在下面的 test_addyu.py 文件中调用 addyu.py 内的 add(a,b)函数。test_addyu.py 文件的代码如下：

```
#test_addyu.py
from addyu import add
a=add(1,2)
```

这里要注意 addyu.py 保存的位置。为了让 Python 解释器能直接 import 默认安装路径以外的第三方模块(如我们自行编写的模块)，需要在系统环境中添加第三方模块路径，即新建 pythonpath 环境变量，值为这个模块所在的路径。具体方法如下。

打开计算机“控制面板”，选择“系统和安全”，再选择“系统”，单击“高级系统设置”，弹出如图 3-4 所示的界面。

单击“环境变量”按钮，在弹出的界面中，在“系统变量”下单击“新建”按钮，在弹出的“新建系统变量”对话框中，为变量名填写“pythonpath”，为变量值填写第三方模块文件所在的路径即可。

如这里存储路径填写了“D:\python_yubg”，如图 3-5 所示。

也可以用临时访问文件的方法，如调用在 E:/yubg/python 中的文件 yubg.py：

```
import sys
sys.path.append('E:/yubg/python')
import yubg
```



图 3-4　新建 pythonpath 环境变量

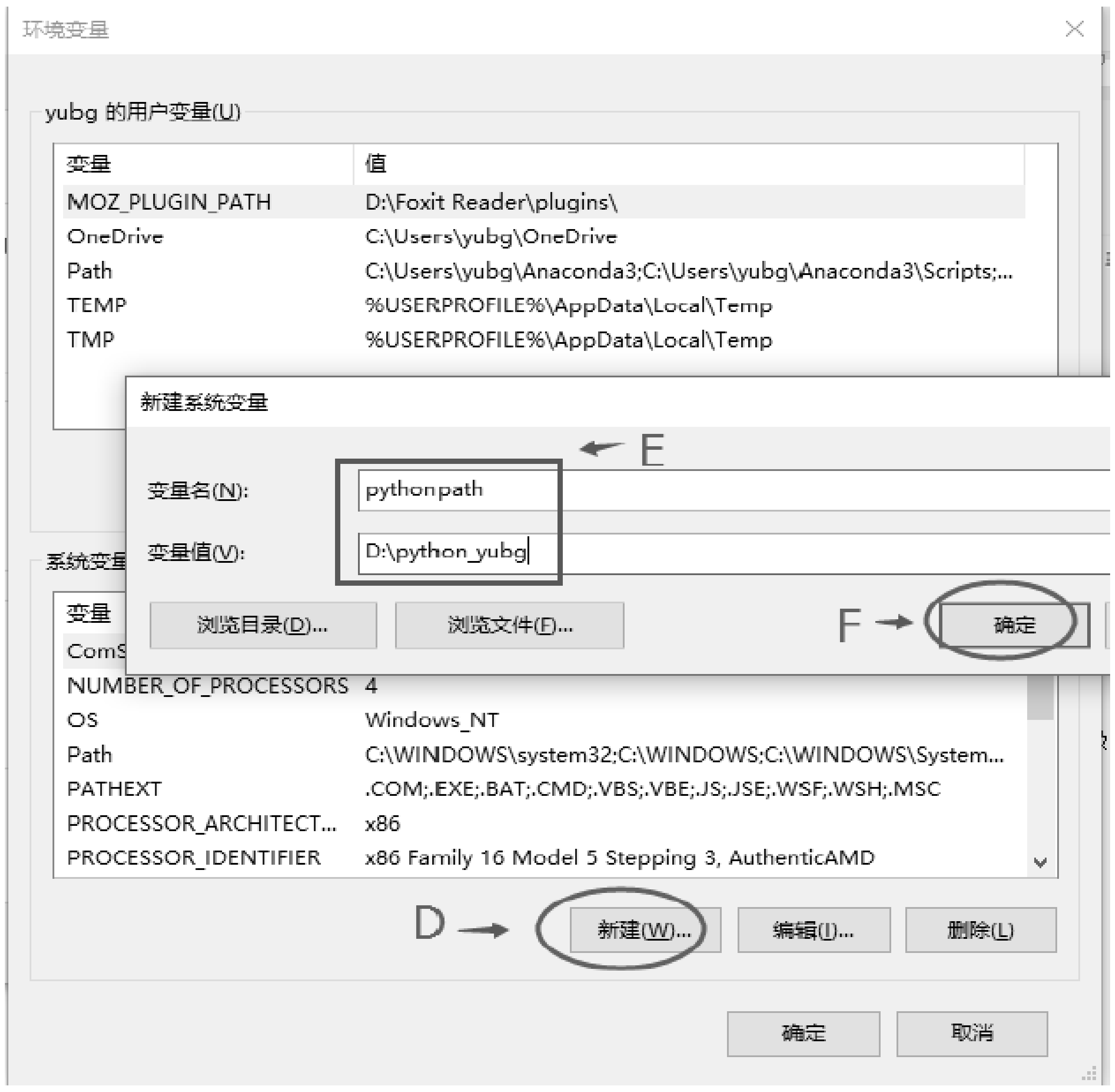


图 3-5　存储路径

3.7.2 包(package)

Python 包是一个有层次的文件目录结构，它定义了由 n 个模块或 n 个子包组成的 Python 应用程序执行环境。

简单地说，包是一个包含__init__.py 文件的目录，该目录下一定得有__init__.py 文件和其他模块或子包，也就是带有__init__.py 的文件夹，并不在乎里面有什么。

多个关系密切的模块组织成一个包，以便于维护和使用。这项技术能有效避免名字空间冲突。创建一个名为包名的文件夹，并在该文件夹下创建一个__init__.py 文件，就定义了一个包。可以根据需要，在该文件夹下存放资源文件、已编译扩展及子包。

举例来说，一个包可能有以下结构：

```
yubg/
    __init__.py
    index.py
    Primitive/
        __init__.py
        lines.py
        fill.py
        text.py
        ...
    yubg_1/
        __init__.py
        plot2d.py
        ...
    yubg2/
        __init__.py
        plot3d.py
        ...
```

import 语句使用以下几种方式导入包中的模块：

```
import yubg.Primitive.fill
#导入模块 yubg.Primitive.fill，只能以全名访问模块属性
#例如 yubg.Primitive.fill.floodfill(img,x,y,color)

from yubg.Primitive import fill
#导入模块 fill，只能以 fill.属性名 这种方式访问模块属性
#例如 fill.floodfill(img,x,y,color)

from yubg.Primitive.fill import floodfill
#导入 fill，并将函数 floodfill 放入当前名称空间，直接访问被导入的属性
#例如 floodfill(img,x,y,color)
```

无论一个包的哪个部分被导入，在文件__init__.py 中的代码都会运行。这个文件的内容允许为空，不过，通常情况下，它用来存放包的初始化代码。导入过程遇到的所有__init__.py 文件都被运行。因此，import yubg.Primitive.fill 语句会顺序运行 yubg 和 Primitive 文件夹下的__init__.py 文件。

下面的语句有歧义：

```
from yubg.Primitive import *
```

本语句的原意是想将 yubg.Primitive 包下的所有模块导入到当前的名称空间。然而，由于不同平台之间文件命名规则不同(比如大小写敏感问题)，Python 不能正确判断哪些模块要被导入。语句只会顺序运行 yubg 和 Primitive 文件夹下的__init__.py 文件。要解决这个问题，应该在 Primitive 文件夹下的__init__.py 中，定义一个名字为 all 的列表，例如：

```
# yubg/Primitive/__init__.py
__all__ = ["lines","text","fill",...]
```

这样，上面的语句就可以导入列表中所有的模块了。

3.7.3　datetime 和 calendar 模块

time 模块提供各种时间相关的功能。在 Python 中，与时间处理有关的模块包括 time、datetime，以及 calendar 等。

(1)　必要说明。

虽然 time 模块总是可用，但并非所有的功能都适用于各个平台。模块中定义的大部分函数是调用 C 平台上的同名函数实现，所以各个平台上的实现可能略有不同。

(2)　一些术语和约定的解释。

时间戳(timestamp)的方式：通常来说，时间戳表示的是从 1970 年 1 月 1 日 00:00:00 开始按秒计算的偏移量(time.gmtime(0))，此模块中的函数无法处理 1970 纪元年以前的日期和时间或太遥远的未来(处理极限取决于 C 函数库，对于 32 位系统来说，是 2038 年)。

UTC(Coordinated Universal Time，世界协调时)：也叫格林尼治天文时间，是世界标准时间。在中国为 UTC+8。

DST(Daylight Saving Time)：即夏令时的意思。

【例 3-14】time 模块：

```
>>> import time                #导入时间模块
>>> t1=time.time()             #返回现在的时间，但返回的是时间戳
>>> t1
1466828024.2864037
>>>
```

```
>>> t2=time.ctime()          #返回现在的时间，正常的时间格式
>>> t2
'Sat Jun 25 12:14:25 2016'
>>>

>>> t2=time.ctime(t1)    #可以将时间戳作为参数，返回正常时间格式
>>> t2
'Sat Jun 25 12:13:44 2016'
>>>

>>> t3=time.localtime()    #返回当前的时间元组格式
>>> t3
time.struct_time(tm_year=2016, tm_mon=6, tm_mday=25, tm_hour=12,
tm_min=15, tm_sec=57, tm_wday=5, tm_yday=177, tm_isdst=0)
>>>

>>> t4=time.asctime()          #返回现在的时间，正常的时间格式
>>> t4
'Sat Jun 25 12:16:35 2016'
>>>

>>> t4=time.asctime(t3)     #可将时间元组作为参数，返回正常的时间格式
>>> t4
'Sat Jun 25 12:15:57 2016'
>>>

>>> time.strftime('%y/%m/%d') #返回当前日期，以/分割，也可换成以,分割
'16/06/25'
>>>
```

说明如下。

tm_year：年。

tm_mon：月。

tm_mday：日。

tm_hour：时。

tm_min：分。

tm_sec：秒。

tm_wday：一周中的第几天。

tm_yday：一年中的第几天。

tm_isdst：夏令时(−1 代表夏令时)。

1．datetime 模块

datatime 模块重新封装了 time 模块，提供更多接口，提供的类有 date、time、datetime、timedelta(时间加减)、tzinfo(时区)。

例如：

```
>>> import datetime
>>> datetime.date.today()             #返回当前日期
datetime.date(2016, 6, 25)

>>> datetime.date(2016, 4, 10)
datetime.date(2016, 4, 10)

>>> datetime.date.today().ctime()      #返回当前日期时间
'Sat Jun 25 00:00:00 2016'

>>> datetime.date.today().timetuple()  #返回当前的时间元组格式
time.struct_time(tm_year=2016, tm_mon=6, tm_mday=25, tm_hour=0, tm_min=0,
tm_sec=0, tm_wday=5, tm_yday=177, tm_isdst=-1)

>>> print(datetime.datetime.now())    #返回当前正常的日期时间
2016-06-25 13:56:47.281702

>>> t = datetime.datetime.now()      #取当前日期时间
>>> m = t + datetime.timedelta(5)  #在 t 时刻上增加 5 天(默认是天)
>>> m
datetime.datetime(2016, 6, 30, 13, 56, 53, 998939)

>>> m = t + datetime.timedelta(weeks=5)
>>> print(m)
2016-07-30 13:56:53.998939
>>>
```

说明如下。

timedelta()的参数还可以是 hours=5，或者是 weeks=5、minutes=5、seconds=5，但没有 years 和 months，因为年和月的时间不定，如月份有 30 天和 31 天。

2．calendar 模块

此模块的函数都是与日历相关的，例如打印某月的字符月历。星期一是默认的每周第一天，星期天是默认的最后一天。更改设置需调用 calendar.setfirstweekday()函数。

例如：

```
>>> import calendar
>>> m = calendar.month(2016,4)    #返回某年某月的日历
>>> print(m)
    April 2016
Mo Tu We Th Fr Sa Su
             1  2  3
4   5  6   7  8   9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30

>>>
print(n)

>>> n = calendar.calendar(2016,w=2,l=1,c=6)   #见后注
>>> print(n)
```

```
                                  2016

      January                   February                   March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
             1  2  3       1  2  3  4  5  6  7          1  2  3  4  5  6
 4  5  6  7  8  9 10       8  9 10 11 12 13 14       7  8  9 10 11 12 13
11 12 13 14 15 16 17      15 16 17 18 19 20 21      14 15 16 17 18 19 20
18 19 20 21 22 23 24      22 23 24 25 26 27 28      21 22 23 24 25 26 27
25 26 27 28 29 30 31      29                        28 29 30 31

       April                      May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
             1  2  3                         1             1  2  3  4  5
 4  5  6  7  8  9 10       2  3  4  5  6  7  8       6  7  8  9 10 11 12
11 12 13 14 15 16 17       9 10 11 12 13 14 15      13 14 15 16 17 18 19
18 19 20 21 22 23 24      16 17 18 19 20 21 22      20 21 22 23 24 25 26
25 26 27 28 29 30         23 24 25 26 27 28 29      27 28 29 30
                          30 31

        July                     August                  September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
             1  2  3       1  2  3  4  5  6  7                1  2  3  4
 4  5  6  7  8  9 10       8  9 10 11 12 13 14       5  6  7  8  9 10 11
11 12 13 14 15 16 17      15 16 17 18 19 20 21      12 13 14 15 16 17 18
18 19 20 21 22 23 24      22 23 24 25 26 27 28      19 20 21 22 23 24 25
25 26 27 28 29 30 31      29 30 31                  26 27 28 29 30

      October                   November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
                1  2          1  2  3  4  5  6                1  2  3  4
 3  4  5  6  7  8  9       7  8  9 10 11 12 13       5  6  7  8  9 10 11
10 11 12 13 14 15 16      14 15 16 17 18 19 20      12 13 14 15 16 17 18
17 18 19 20 21 22 23      21 22 23 24 25 26 27      19 20 21 22 23 24 25
24 25 26 27 28 29 30      28 29 30                  26 27 28 29 30 31
31

>>>
```

注：n = calendar.calendar(2016,w=2,l=1,c=6)——返回某年的年历，三个月一行，间隔为距离 c，每日宽度间隔为 w 字符，每行长度为 21×w+18+2×c，l 是每星期行间距。

calendar 模块还可以处理闰年的问题。判断是否闰年、两个年份之间闰年的个数。

例如：

```
>>> import calendar
>>> print(calendar.isleap(2012))
True
>>> print(calendar.leapdays(2010, 2015))
1
>>>
```

3.7.4 urllib 模块

当我们想下载网上的文档、数据时，也就是编写大家常说的爬虫，常用到 urllib 模块，具体示例如下：

```
>>> import urllib.request
>>> ur = urllib.request.urlopen("http://www.baidu.com")
>>> content = ur.read()
>>> mystr = content.decode("utf8")
>>> ur.close()
>>> print(mystr)

<!DOCTYPE html>
<!--STATUS OK-->
<html>
<head>
    <meta http-equiv="content-type" content="text/html;charset=utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=Edge">
     <meta content="always" name="referrer">
    <meta name="theme-color" content="#2932e1">
    <link rel="shortcut icon" href="/favicon.ico" type="image/x-icon" />
    <link rel="search" type="application/opensearchdescription+xml"
href="/content-search.xml" title="百度搜索" />
    <link rel="icon" sizes="any" mask
href="//www.baidu.com/img/baidu.svg">

     <link rel="dns-prefetch" href="//s1.bdstatic.com"/>
     <link rel="dns-prefetch" href="//t1.baidu.com"/>
     <link rel="dns-prefetch" href="//t2.baidu.com"/>
     <link rel="dns-prefetch" href="//t3.baidu.com"/>
```

```
    <link rel="dns-prefetch" href="//t10.baidu.com"/>
    <link rel="dns-prefetch" href="//t11.baidu.com"/>
    <link rel="dns-prefetch" href="//t12.baidu.com"/>
    <link rel="dns-prefetch" href="//b1.bdstatic.com"/>

   <title>百度一下，你就知道</title>
   ...
>>>
```

由于网页内容太多，这里只取了<title>标签之前的部分。当然，要想获取更多的其他内容，那就要使用更多的技术。这里仅仅把首页上的内容“抓”了下来。需要注意：

```
urllib.request.urlopen("http://www.baidu.com")
```

这行代码里的参数是网址，“http://”不能少，否则会报错！关于更多的更深层次的爬虫技术，本书后面还有介绍。

3.8 类

面向对象编程——Object Oriented Programming，简称 OOP，是一种程序设计思想。OOP 把对象作为程序的基本单元，一个对象包含了数据和操作数据的函数。

在 Python 中，所有数据类型都可以视为对象，当然，也可以自定义对象。自定义的对象数据类型就是面向对象中的类(Class)的概念。

类是所有面向对象语言中最难理解的一个内容。Python 中的类(Class)是一个抽象的概念，比函数还要抽象，这也是 Python 的核心概念，是一个非常重要的知识点，我们可以把它简单地看作是数据以及由存取、操作这些数据的方法所组成的一个集合。

我们在学习函数(function)之后，知道了可以重用代码，那为什么还要用类来规划函数呢？因为类有这样一些优点。

(1) 类对象是多态的：也就是多种形态，这意味着我们可以对不同的类对象使用同样的操作方法，而不需要额外写代码。

(2) 类的封装：封装之后，可以直接调用类的对象，来操作内部的一些类方法，不需要让使用者看到代码工作的细节。

(3) 类的继承：类可以从其他类或者元类中继承它们的方法，直接使用。

定义类(class)的语法如下：

```
class NameClass(object):
   def fname(self, name):
       self.name = name
```

第一行语法是 class 后面紧接类的名称，最后带上冒号。类的名称首字母最好大写。

第二行开始是类的方法，与函数非常相似，但是与普通函数不同的是，它的内部有一个 self 参数，作用是对对象自身的引用。

这里以一个例子来说明面向过程和面向对象在程序流程上的不同之处。假设要处理学生的成绩表，为了表示一个学生的成绩，面向过程的程序可以用一个字典来表示：

```
std1 = {'name': 'Michael', 'score': 98}
std2 = {'name': 'Bob', 'score': 81}
```

处理学生成绩可以通过函数来实现，比如打印学生的成绩：

```
def print_score(std):
    print('{0},{1}'.format(std['name'], std['score']))
```

但如果采用面向对象的程序设计思想，我们首先思考的不是程序的执行流程，而是 Student 这种数据类型应该被视为一个对象，这个对象拥有 name 和 score 这两个属性(Property)。如果要打印一个学生的成绩，首先必须创建出这个学生对应的对象，然后，给对象发一个 print_score 消息，让对象自行把自己的数据打印出来。

类定义如下：

```
class Student(object):
    def __init__(self, name, score):
        self.name = name
        self.score = score
    def print_score(self):
        print('{0},{1}'.format(self.name,self.score))
```

给对象发消息，实际上就是调用对象对应的关联函数，又称为对象的方法(Method)。面向对象的程序写出来就像这样：

```
bart = Student('Bart Simpson', 59)
lisa = Student('Lisa Simpson', 87)
bart.print_score()
lisa.print_score()
```

面向对象的设计思想是从自然界中来的，因为在自然界中，类(Class)和实例(Instance)的概念是很自然的。Class 是一种抽象概念，比如我们定义的 Class——Student，是指学生这个概念，而实例(Instance)则是一个个具体的 Student，比如 Bart Simpson 和 Lisa Simpson 是两个具体的 Student。

所以，面向对象的设计思想是抽象出 Class，根据 Class 创建 Instance。

类的方法与普通的函数只有一个特殊的区别——它们必须有**额外的第一个参数**，但是

在调用此方法的时候，不用为这个参数赋值，Python 会提供这个值。这个特别的变量指对象本身，它就是 self。

虽然可以给这个参数任何名称，但是强烈建议使用 self 这个名称——其他名称都是不赞成使用的。使用一个标准的名称有很多优点，例如我们写的程序代码其他人可以迅速识别，如果使用 self 的话，还有些 IDLE(集成开发环境)也可以帮助我们。

Python 如何给 self 赋值以及为何不需要给它赋值？这里举一个例子。

假如有一个 MyClass 类和它的一个实例 MyObj。调用对象的 MyObj.method(arg1,arg2)方法时，Python 会自动转为 MyClass.method(MyObj, arg1, arg2)，这就是 self 的原理了。

这也意味着如果有一个不需要参数的方法，但是我们仍然需要给这个方法定义一个 self 参数。比较下面类和函数的区别。

类：

```
#coding = utf-8
#创建实例类 Test
class Test(object):
    def add(self,a,b):
        '''
        打印输出 a+b 的值，注意类方法需要添加 self
        '''
        print(a+b)
    def display(self):
        '''
        打印输出字符串，注意类方法需要添加 self，尽管不需要参数
        '''
        print('hell, here is a Class test.')

test = Test()     #创建一个类实例 test
test.add(1,3)     #调用方法
test.display()
```

函数：

```
def addTwo(a,b):   #不需要加 self 参数
    print(a+b)

addTwo(1,2)
```

类中声明 add()方法时，若不加 self，则提示错误。

Python 编程中，类的概念可以比作某种类型集合的描述。打个比方：类就是烤饼干的模子，而一个个的饼干就是一个个实例，或者说类就是一个工厂，实例就是一个个产品。

创建类时，可以定义一个特定的方法，名为__init__()，只要创建这个类的一个实例，就会运行这个方法。可以向__init__()方法传递参数，这样，创建对象时就可以把属性设置为我们希望的值，这个__init__()方法会在创建对象时完成初始化。

```
>>> class peo:
    def __init__(self,name,age,sex):
        self.Name = name
        self.Age = age
        self.Sex = sex
    def speak(self):
        print("my name: " + self.Name)

>>>
```

实例化这个类的对象时：

```
>>> zhangsan=peo("zhangsan",24,'man')
>>> print(zhangsan.Age)
 24
>>> print(zhangsan.Name)
 zhangsan
>>> print(zhangsan.Sex)
 Man
>>> zhangsan.speak()
my name: zhangsan
>>>
```

本 章 小 结

本章主要学习了 if、for、while 流程控制语句，尤其使用 for 循环遍历的方法，还有以下重要内容。

(1) 函数 map 和 zip 的使用。

(2) 函数和类的编写格式。

(3) try 的使用方法。

(4) 包和模块的导入方法。

练　　习

(1) 在 0~9 之间随机选择 1 个整数，操作 100 次，统计共有几种数字，并用字典的方式输出每个数字的出现次数，键是出现的整数，值是出现的次数。

(2) 将整数 2016 的每个数字分离出来，依次打印输出。

(3) 已知字典{"name": "python", "book": "python", "lang": "english"}，要求将该字典的键和值对换。(注意，字典中有键的值是重复的)

(4) 已知一段程序中，用列表保存几个用户名，例如['xiaoxifeng', 'cangcang', 'tom']，要求通过终端输入新的用户名，判断所输入的用户名是否为已经设置好的用户名，并且对判断结果给出友好的提示。如果不是，允许用户多次尝试输入，直到正确为止。

(5) 找一段英文的文本，统计该文本中单词的出现次数。比如"How are you. How are you."的统计结果是{"how":2, "are":2, "you":2}。

(6) 已知字符串 a = "aAsmr3idd4bgs7Dlsf9eAF"，要求编写程序，完成如下任务。

① 将字符串中的数字取出，并输出成一个新的字符串。

② 统计字符串中每个字母的出现次数(忽略大小写，即认为 a 与 A 是同一个字母)，并输出成一个字典。像{'a':3,'b':1}这样。

③ 去除字符串多次出现的字母，不区分大小写。如'aAsmr3idd4bgs7Dlsf9eAF'经过去除后，输出'asmr3id4bg7lf9e'。

(7) 有一百个瓶子，分别编号为 1~100。现在有人拿枪从第一个开始射击，每枪击破一个，跳过一个，一直到一轮完成。接着在剩下的瓶子里面再次击破第一个，间隔一个再击破一个。问最后剩下完整的瓶子是这一百个瓶子里的第几个？

(8) 写一段程序，能够实现如下功能。

① 输入英文的姓名。

② 按照字典顺序将所有姓名排序。

③ 输入完毕，将排序结果打印出来。

(9) 编写一个函数，实现摄氏温度和华氏温度之间的换算，换算公式为 F=9C/5 + 32。要求输入单位是摄氏度的值，能够显示相应的华氏度值，反之亦然。

(10) 制做一个加法计算器，要求用户先后输入两个数，能够计算出结果，并打印出加法算式。

(11) 为老师们编写一个处理全班考试成绩的程序。要求如下。

① 能够依次录入班级同学的姓名和分数。

② 录入完毕，则打印出全班的平均分、最高分的同学姓名和分数。

(12) 编写工资额计算器，要求如下。

① 确定每月的基本工资。

② 输入每月的实际应当工作天数。

③ 输入当月的请假天数，如果请假天数小于等于 2 天，对工资无影响；大于 2 天小于等于 7 天，扣除当月基本工资的 10%；大于 7 天小于等于 14 天，扣除当月基本工资的

50%；大于 14 天，扣除全月工资。

④　如果当月实际工作天数和应工作天数一样(不算加班)，则增加基本工资的 20%。

⑤　如果当月有加班，则按照加班的天数和当月的日工资(基本工资/实际工作天数)计算加班费。

⑥　输入最终应得工资。

(13) 有多少个三位数能被 17 整除？编写程序，将这些数值显示出来。

(14) 编写一个猜数游戏，要求如下。

①　用户可以输入无限多次数字。

②　如果猜中了数字，则要输出用户猜测的次数和数字结果。

(15) 编写程序，判断一个数是否为素数。

(16) 创建 PayCalculator 类，拥有 pay_rate 属性，以每小时人民币数量为单位。该类拥有 compute_pay(hours)方法，计算给给定工作时间的报酬，并返回。

(17) 创建 SchoolKid 类，初始化小孩的姓名、年龄。也有访问每个属性的方法和修改属性的方法。然后创建 ExaggeratingKid 类，继承 SchoolKid 类，在子类中覆盖访问年龄的方法，并将实际年龄加 2。

第 4 章

Python 数据分析实战

本章主要介绍 Python 在数据处理、数据分析、数据可视化方面的常用方法与技巧。

4.1 关于 Pandas

4.1.1 什么是 Pandas

Pandas 是 Python 的一个数据分析包，最初由 AQR Capital Management 于 2008 年 4 月开发，并于 2009 年底开源面市，目前由专注于 Python 数据包开发的 PyData 开发团队继续开发和维护，属于 PyData 项目的一部分。

Pandas 最初被作为金融数据分析工具而开发出来，因此，Pandas 为时间序列分析提供了很好的支持。Pandas 的名称来自于面板数据(panel data)和 Python 数据分析(data analysis)。panel data 是经济学中关于多维数据集的一个术语，在 Pandas 中，也提供了 panel 的数据类型。

4.1.2 Pandas 中的数据结构

Pandas 中引入了两种新的数据结构——Series 和 DataFrame，这两种数据结构都建立在 NumPy 的基础之上。

Series：一维数组系列，也有称序列的，与 Numpy 中的一维 array 类似。二者与 Python 基本的数据结构 List 也很相近。

DataFrame：二维的表格型数据结构。很多功能与 R 中的 data.frame 类似。可以将 DataFrame 理解为 Series 的容器。以下内容主要以 DataFrame 为主。

Panel：三维的数组，可以理解为 DataFrame 的容器。

4.1.3 Pandas 的安装方法

在安装 Pandas 之前，无须安装任何 Python 及其衍生产品，直接下载 Anaconda，官方网址为 https://www.continuum.io/downloads，Anaconda 发展更新较快，本书下载的是 Python 3.5 版本，32 位(看个人机器，有的是 64 位)。

下载界面如图 4-1 所示。

下载后的文件如下：

Anaconda3-2.4.1-Windows-x86.exe	应用程序	306,282 KB

直接双击安装，可自选安装位置。安装完成后，在开始菜单里可以看到如图 4-2 所示的菜单。

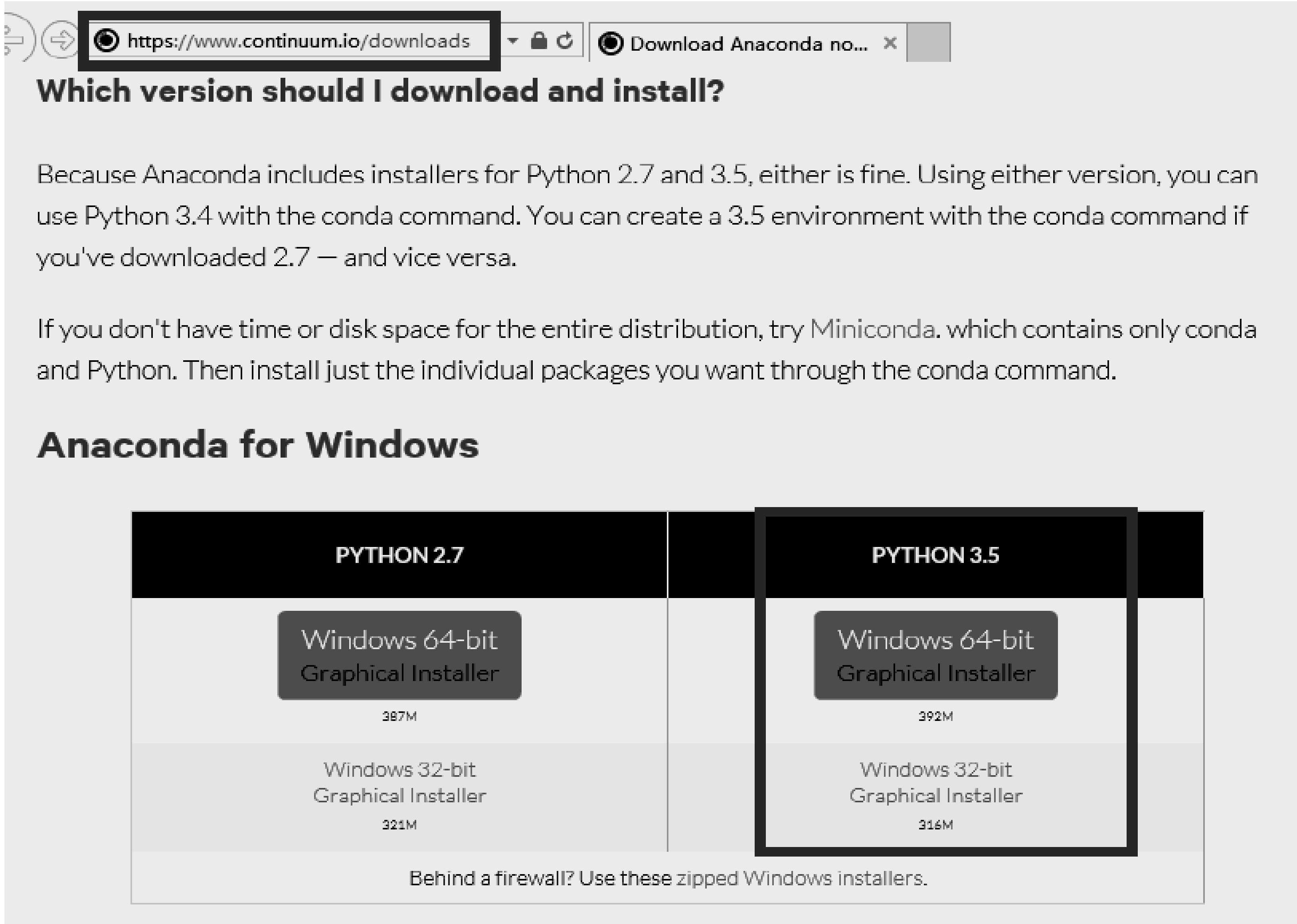


图 4-1　Anaconda 官网下载界面

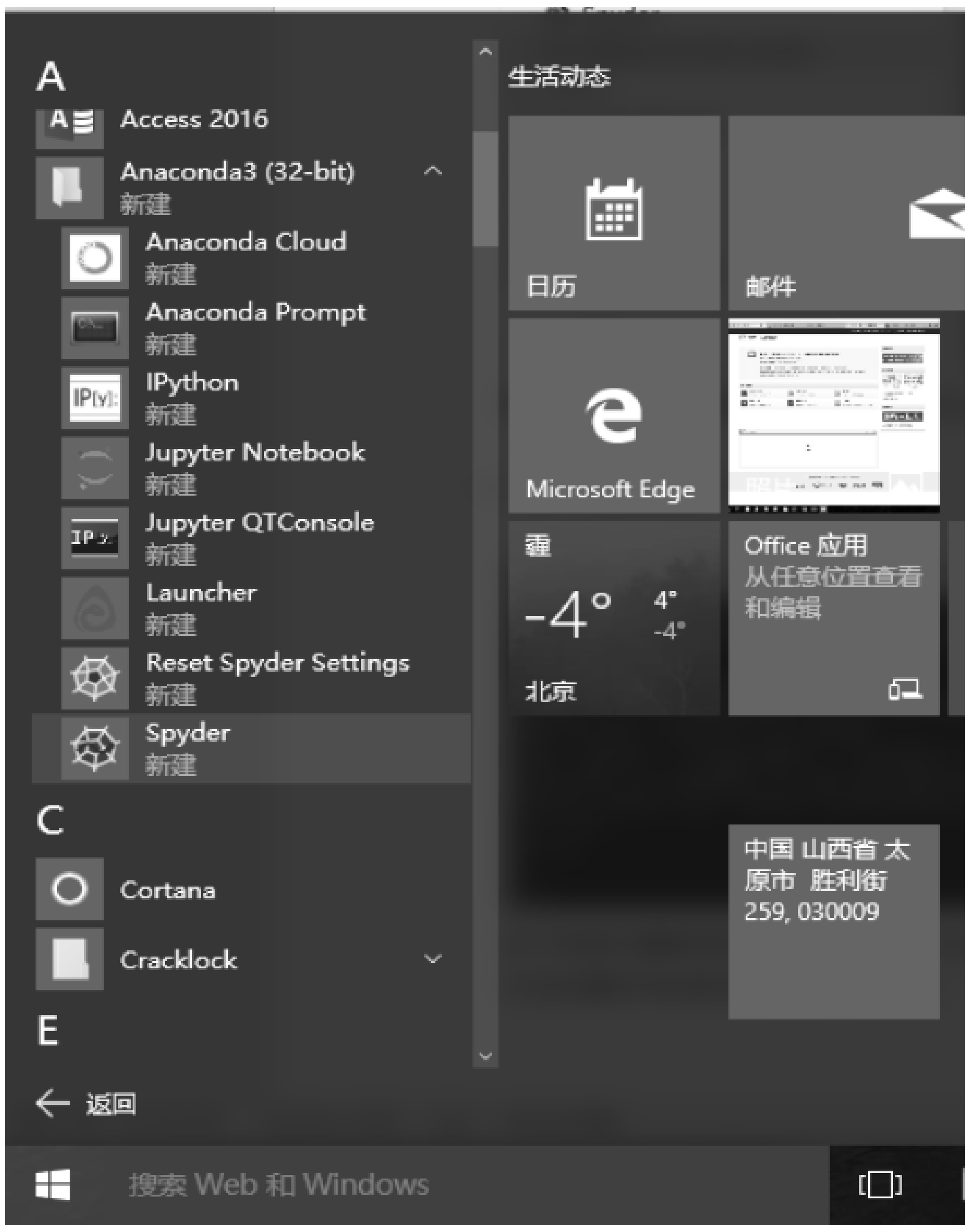


图 4-2　Anaconda 菜单

安装完 Anaconda，就相当于安装了 Python、IPython、集成开发环境 Spyder 以及一些安装包等。

应注意：Windows 7 下 64 位安装一样进行，但有些机器安装完毕后，在开始菜单栏内找不到 Spyder，但可以运行 anaconda prompt，再键入 Spyder 即可运行。

Spyder 是使用 Python 语言进行科学编程的跨平台开源集成开发环境，它的最大优点就是模仿 MATLAB 的“工作空间”，第一次打开比较慢。Spyder 的使用比较简单，下面介绍它的几个基本功能。

1. 代码提示

代码提示是开发工具必备的功能，当需要 Spyder 给我们进行代码提示时，只需要输入函数名的前几个字母，再按下 Tab 键，即可得到 IDE 的代码提示，如图 4-3 所示。

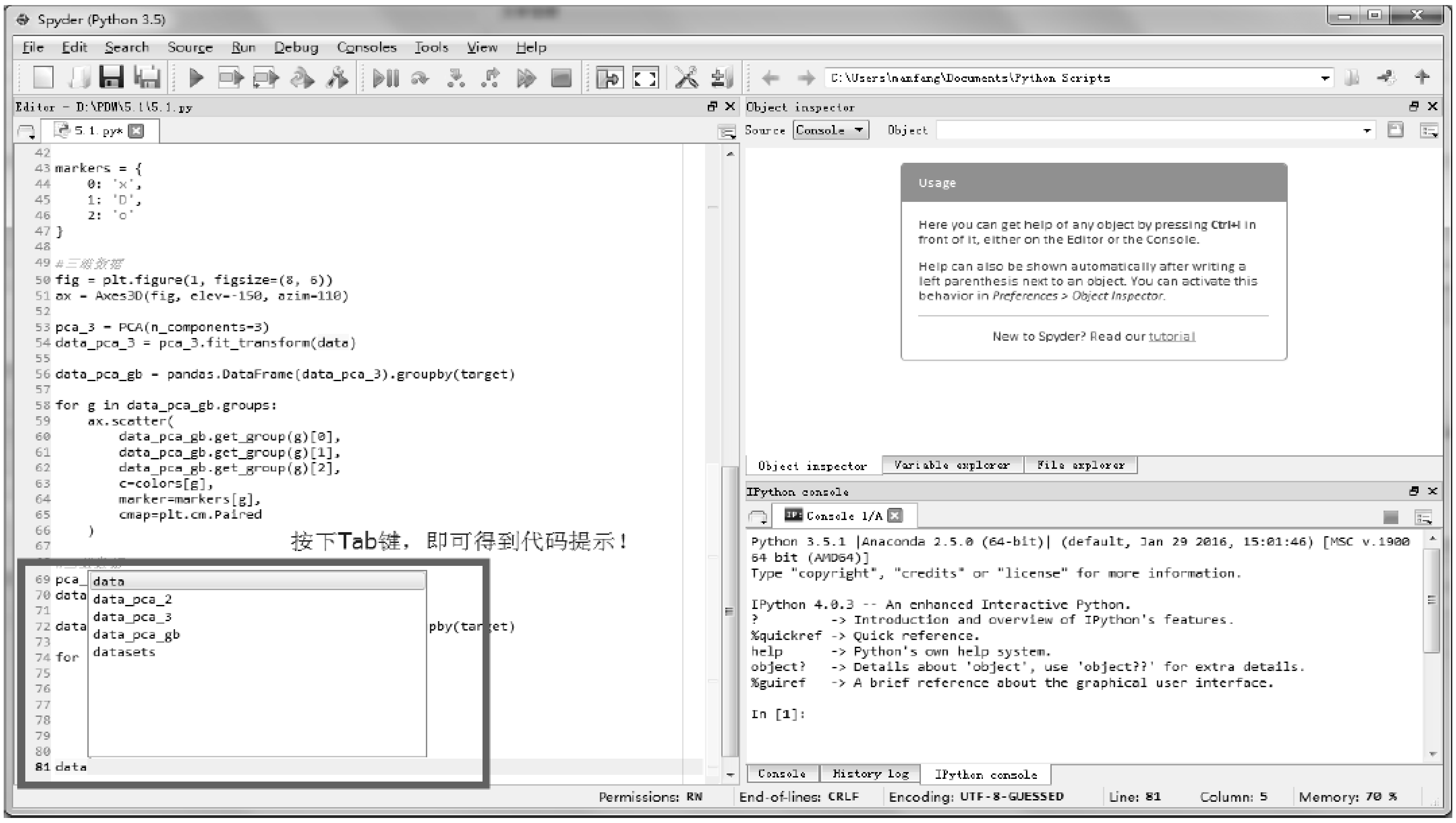


图 4-3 Spyder 界面

2. 变量浏览

变量是代码执行过程中暂留在内存中的数据，我们可以通过 Spyder 对变量承载的数据进行查看，方便我们对数据进行处理。

如图 4-4 所示，变量浏览框中包含了变量的名称、类型、大小以及基本预览，双击对应变量名所在的行，即可打开变量的详细数据进行查看。

3. 图形查看

绘图是我们进行数据分析时必备的技能之一，一款好的工具，必须具备图形绘制的功能，IPython 窗体中集成了绘图功能，如图 4-5 所示。

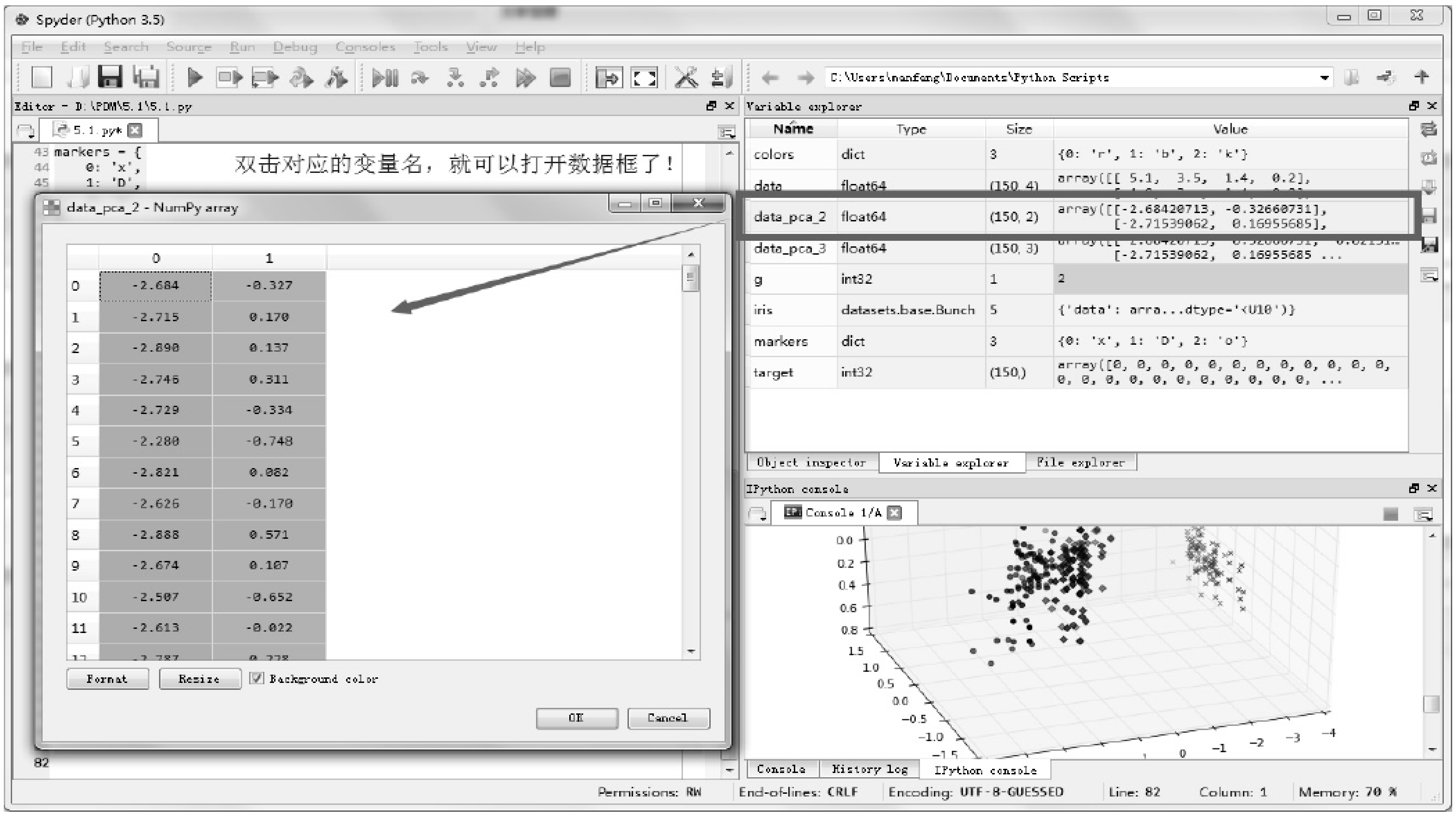


图 4-4　Spyder 变量浏览

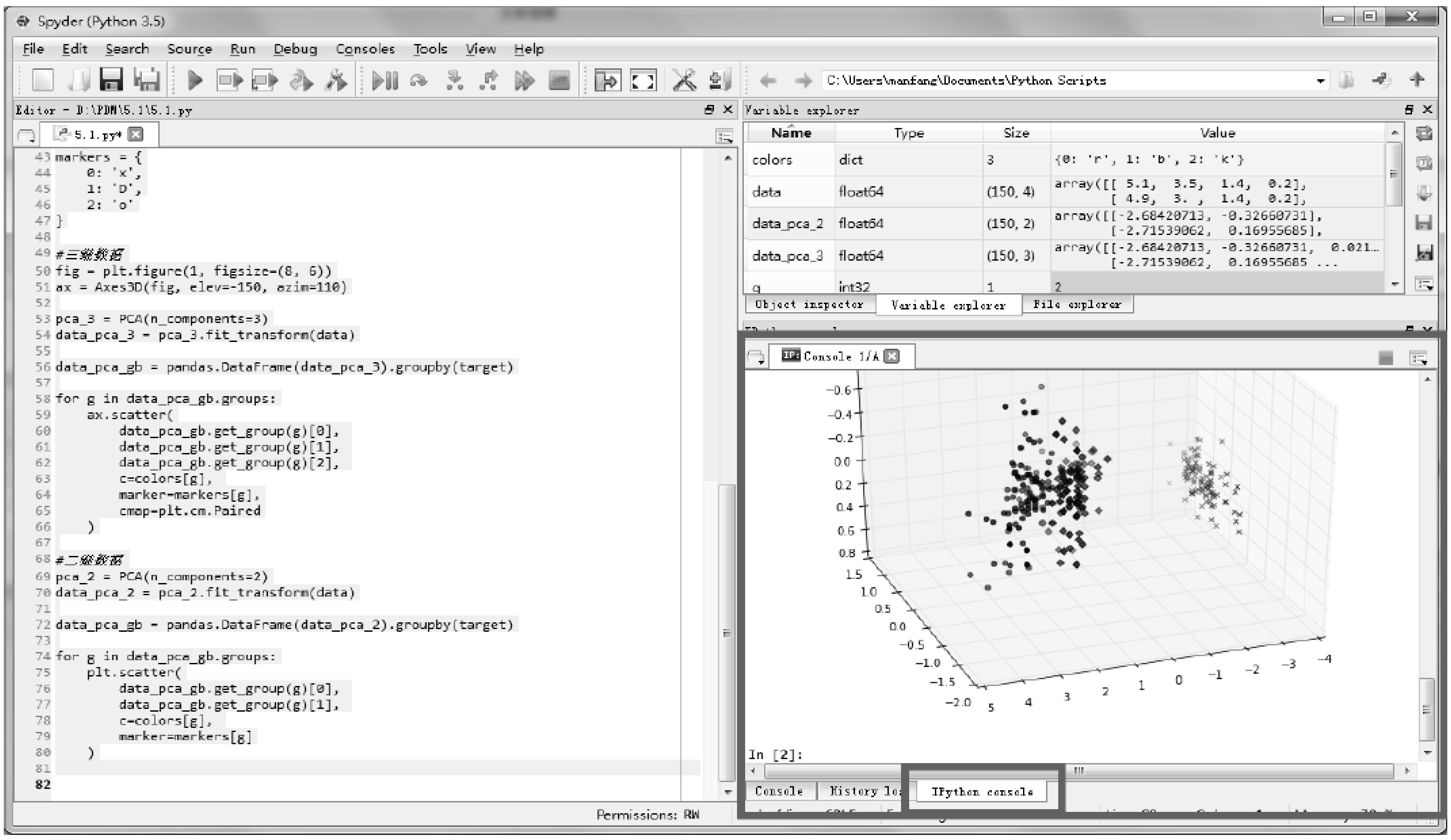


图 4-5　Spyder 绘图界面

了解以上三点功能，基本上就可以使用 Spyder 在数据分析过程中尽情发挥了。

最后，需要提醒大家的是，执行代码时必须先选中代码，然后再按下 Ctrl+Enter 组合键，如果没有选中代码，只是把光标放在代码对应的行，按下 Ctrl+Enter 组合键，是不能

执行代码行的。执行选定的代码也可以用鼠标单击 Run cell 按钮来完成，如图 4-6 所示。



图 4-6　单击 Run cell 按钮执行选定的代码

如果在安装时遇到其他问题，可以在如下网址中找到各种操作系统的详细安装指导：

```
http://www.datarobot.com/blog/getting-up-and-running-with-python/
```

4.1.4　在 Anaconda 中安装第三方库

在 Windows 下安装 Python 和很多包，对于新手来说，总觉得是一件很痛苦的事情。所以许多人喜欢使用 Anaconda，最方便的一点是它整合了大量的依赖包。

如下网址页面列出了它所包含的全部依赖包：

```
http://docs.continuum.io/anaconda/pkg-docs.html
```

其中，比如科学计算的 numpy、theano 等，都应有尽有。

尽管 Anaconda 整合了很多常用的包，但它也不是万能的，有些包就没有整合进来，比如爬虫 scrapy。

不过安装 scrapy 包比较简单，只需要在“开始”菜单中选定“Windows 系统”，单击“命令提示符”，在弹出的窗口命令行中输入“conda install scrapy”，就可以安装了。

图 4-7 是安装 scrapy 包的截图。

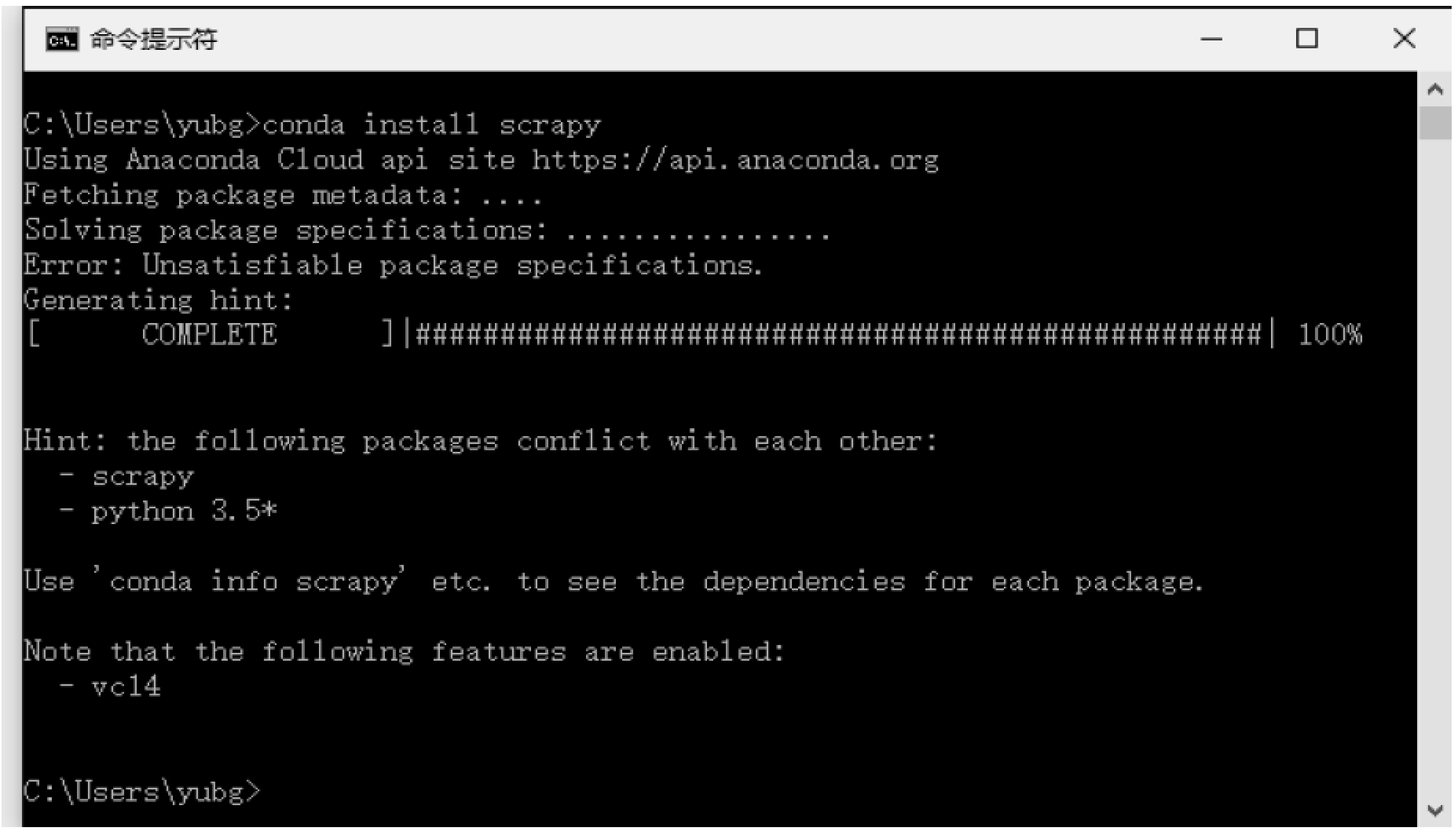


图 4-7　安装 scrapy 包的界面

4.2　数 据 准 备

4.2.1　数据类型

Python 常用的三种数据类型是 logical、numeric、character。

logical 即布尔型，只有两种取值，0 和 1，或者真假(true、false)。

运算规则：&(与，有一个为假则为假)；|(或，有一个为真则为真)；not(非，取反)，具体如表 4-1 所示。

表 4-1　布尔运算规则

运 算 符	注　释	运算规则
&	与	两个逻辑型数据中，其中一个数据为假，则结果为假
|	或	两个逻辑型数据中，其中一个数据为真，则结果为真
not	非	取相反值，非真的逻辑型数据为假，非假的逻辑型数据为真

numeric 即数值型。

运算规则：+、-、*、/ 。

character 即字符型，使用单引号(‘’)或者双引号(“”)包围起来。

Python 数据类型变量的命名规则如下。

(1)　变量名可以由 a~z、A~Z、数字、下划线组成，首字母不能是数字和下划线。

(2) 大小写敏感。

(3) 变量名不能为 Python 中的保留字。如不能是 and、continue、lambda、or 等。

4.2.2 数据结构

数据结构是指相互之间存在的一种或多种特定关系的数据类型的集合，主要有 Series (系列)和 Dataframe(数据框)。

1. Series

Series 即系列(也称序列)，用于存储一行或一列的数据，以及与之相关的索引的集合。使用方法如下：

```
Series([数据 1, 数据 2, ...], index=[索引 1, 索引 2, ...])
```

例如：

```
In [1]:from pandas import Series
       X = Series(['a',2,'螃蟹'],index=[1,2,3])
In [1]:X
Out[2]:
1     a
2     2
3    螃蟹
dtype: object

In [3]:X[3]
Out[3]:'螃蟹'
```

一个系列允许存放多种数据类型，索引也可以省略；可以通过位置或者索引访问数据，如 X[3]，返回‘螃蟹’。

Series 的索引 index 可以省略，默认从 0 开始，也可以指定索引。

在 Spyder 中写入代码：

```
from pandas import Series
A=Series([1,2,3])  #定义系列的时候，数据类型不限
print(A)

#输出如下
0    1          #第一列的 0 到 2 就是数据的索引，也就是位置，计数从 0 开始
1    2
2    3
```

```
dtype: int64

from pandas import Series
A=Series([1,2,3],index=[1,2,3])  #可自定义索引，如：123；ABCD 等
print(A)

1    1
2    2
3    3
dtype: int64           #dtype 指向数据类型，ini64 是指 64 位整数

from pandas import Series
A=Series([1,2,3],index=['A','B','C'])
print(A)

A    1
B    2
C    3
dtype: int64
```

注意容易犯的错误：

```
from pandas import Series
A=Series([1,2,3],index=[A,B,C])
print(A)

Traceback (most recent call last):

  File "<ipython-input-10-d5dd51933cbd>", line 3, in <module>
    A=Series([1,2,3],index=[A,B,C])

NameError: name 'B' is not defined
```

因为这里 A、B、C 都是字符串，需要使用引号。

访问系列值时，需要通过索引值来访问，系列的索引 index 和系列值是一一对应的关系，如表 4-2 所示。

表 4-2　系类索引与系列值对应

系列索引	系 列 值
0	14
1	26
2	31

例如：

```
from pandas import Series
A=Series([14,26,31])
print(A)
print(A[1])     #系列的位置从 0 开始的，第一个数从 0 开始计数
print(A[5])     #超出 index 的总长度会报错

0    14
1    26
2    31
dtype: int64
26
KeyError: 5     # print(A[5])时因为索引越界出错

from pandas import Series
A=Series([14,26,31],index=['first','second','third'])
print(A)
print(A['second'])  #如果设置了 index 参数，也可通过参数来访问系列值

first     14
second    26
third     31
dtype: int64
26
```

执行下面的代码，看看运行的结果：

```
from pandas import Series

#可以混合定义一个序列
x = Series(['a', True, 1], index=['first', 'second', 'third'])

#访问
x[1]

#根据 index 访问
x['second']

#不能越界访问
x[3]

#不能追加单个元素，但可以追加系列
x.append('2')
```

```
#追加一个系列
n = Series(['2'])
x.append(n)

#需要使用一个变量来承载变化，即 x.append(n)返回的是一个新序列
x = x.append(n)

#判断值是否存在，数字和逻辑型(True/False)是不需要加引号的
2 in x.values
'2' in x.values

#切片
x[1:3]

#定位获取，这个方法经常用于随机抽样
x[[0, 2, 1]]

#根据 index 删除
x.drop(0)
x.drop('first')

#根据位置删除，返回新的序列
x.drop(x.index[3])

#根据值删除，显示值不等于 2 的系列，即删除 2，返回新序列
x[2!=x.values]

#通过值访问系列号 index
x.index[x.values=='a']

#修改 series 中的 index：可以通过赋值更改，也可以通过 reindex()方法
x.index=[0,1,2,3,4]

#可将字典转化为 Series
s=Series({'a':1,'b':2,'c':3})
```

Series 的 sort_index(ascending=True)方法可以对 index 进行排序操作，ascending 参数用于控制升序或降序，默认为升序。也可使用 reindex()方法重新排序。

在 Series 上调用 reindex 重排数据，使得它符合新的索引，如果索引的值不存在，就引入缺失数据值：

```
#reindex 重排序
obj = Series([4.5, 7.2, -5.3, 3.6], index=['d', 'b', 'a', 'c'])
obj
Out[25]:
d    4.5
b    7.2
a   -5.3
c    3.6
dtype: float64

obj2 = obj.reindex(['a', 'b', 'c', 'd', 'e'])
obj2
Out[26]:
a   -5.3
b    7.2
c    3.6
d    4.5
e    NaN
dtype: float64

obj.reindex(['a', 'b', 'c', 'd', 'e'], fill_value=0)
Out[27]:
a   -5.3
b    7.2
c    3.6
d    4.5
e    0.0
dtype: float64
```

Series 对象本质上是一个 NumPy 的数组，因此 NumPy 的数组处理函数可以直接对 Series 进行处理。但是 Series 除了可以使用位置作为下标存取元素外，还可以使用标签存取元素，这一点与字典相似。每个 Series 对象实际上都由两个数组组成。

- index：它是从 NumPy 数组继承的 index 对象，保存标签信息。
- values：保存值的 NumPy 数组。

注意如下几点。

(1) Series 是一种类似于一维数组(数组：ndarray)的对象。

(2) 它的数据类型没有限制(各种 NumPy 数据类型)。

(3) 它有索引，把索引当作数据的标签(key)看待，类似于字典(只是类似，实质上是数组)。

(4) Series 同时具有数组和字典的功能，因此它也支持一些字典的方法。

2．DataFrame

DataFrame 是用于存储多行和多列的数据集合，是 series 的容器。使用方式如下：

```
Dataframe(columnsMap)
```

例如：

```
df=DataFrame(
    {'age':Series([21,22,23]),'name':Series(['Yubg','John','Jim'])},
    index=[0,1,2])
```

其中的数据行列位置如图 4-8 所示。

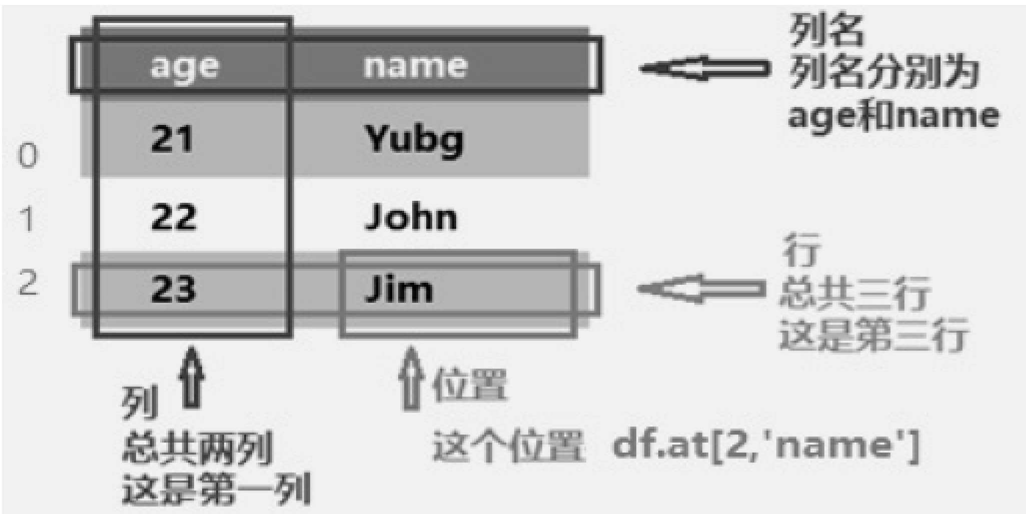


图 4-8　DataFrame 数据行列位置示例

又如：

```
from pandas import Series
from pandas import DataFrame
df=DataFrame({'age':Series([26,29,24]),'name':Series(['Ken','Jerry','Ben
'])})
print(df)

   age   name
0   26    Ken
1   29  Jerry
2   24    Ben
```

注意： DataFrame 中单词的写法是首字母大写。索引不指定时也可以省略。使用数据框时，要先从 pandas 中导入 DataFrame，数据访问方式如表 4-3 所示。

表 4-3　数据框的访问方式

访问位置	方　法	备　注
访问列	变量名[列名]	访问对应的列。如 df[‘name’]
访问行	变量名[n:m]	访问 n 行到 m-1 行的数据。如 df[2:3]

续表

访问位置	方　法	备　注
访问块(行和列)	变量名.iloc[n1:n2, m1:m2]	访问 n1 到(n2-1)行，m1 到(m2-1)列的数据。 如 df.iloc[0:3, 0:2]
访问位置	变量名.at[行名，列名]	访问(行名,列名)位置的数据。如 df.at[1, 'name']

具体示例如下：

```
A=df['age']          #获取 age 的列值
print(A)
0    26
1    29
2    24
Name: age, dtype: int64

B=df[1:2]            #获取序号是第一行的值(其实是第二行，从 0 开始的)
print(B)
   age   name
1   29  Jerry

C=df.iloc[0:2,0:2]  #获取第 0 行到 2 行(不含)与第 0 列到 2 列(不含)的块
print(C)
   age   name
0   26    Ken
1   29  Jerry

D=df.at[0,'name']    #获取第 0 行与 name 列的交叉值
print(D)
Ken
```

执行下面的代码并查看运行结果：

```
from pandas import DataFrame
df1 = DataFrame({
    'age': [21, 22, 23],
    'name': ['KEN', 'John', 'JIMI']});

df2 = DataFrame(data={
    'age': [21, 22, 23],
    'name': ['KEN', 'John', 'JIMI']
}, index=['first', 'second', 'third']);
```

```
#按列访问
df['age']
#按行访问
df1[1:2]

#按行列号访问
df1.iloc[0:1, 0:1]

#按行索引名、列名访问
df2.at["first", 'name']

#修改列名
df1.columns=['age2', 'name2']

#修改行索引
df1.index = range(1,4)

#访问指定列的值
df1[df1.columns[0:2]] #等价于 column_names=df1.columns,df1[column_names[0:2]]

#根据行索引删除
df1.drop(1, axis=0) # axis=0 是表示行轴，也可以省略

#根据列名进行删除
df1.drop('age2', axis=1)  # axis=1 表示列轴，不可省略

#第二种删除列的方法
del df1['age2']

#增加列
df1['newColumn'] = [2, 4, 6]

#增加行。这种方法效率比较低
df2.loc[len(df2)] = [24, "KENKEN"]
```

增加行的办法可以通过合并两个 DataFrame 来解决。例如：

```
df = DataFrame([[1, 2], [3, 4]], columns=list('AB'))
df
Out[46]:
   A  B
0  1  2
1  3  4
```

```
df2 = DataFrame([[5, 6], [7, 8]], columns=list('AB'))
df2
Out[47]:
   A  B
0  5  6
1  7  8

#方法一，合并两个数据框，并生成一个新的数据框，简单的“叠加”，不修改 index
df.append(df2)  #仅把 df 和 df2 “叠”起来了，没有修改合并后 df2 部分的 index
Out[48]:
   A  B
0  1  2
1  3  4
0  5  6
1  7  8

#方法二，合并生成一个新的数据框，并修改 index
df.append(df2, ignore_index=True) #修改 index，对 df2 部分重新索引了
Out[49]:
   A  B
0  1  2
1  3  4
2  5  6
3  7  8
```

4.2.3 数据导入

数据存在的形式多样，有文件(CSV、Excel、TXT)和数据库(MySQL、Access、SQL Server)等形式。

(1) 导入 TXT 文件：

```
read_table(file,names=[列名 1,列名 2,...],sep="",...)
```

file 为文件路径与文件名。

names 为列名，默认为文件中的第一行作为列名。

sep 为分隔符，默认为空，表示默认导入为一列。

【例 4-1】读取(导入)TXT 文件：

```
from pandas import read_table
df = read_table('E://rz1.txt',
names=['YHM','DLSJ','TCSJ','YWXT','IP','REMARK'],sep=",")
print(df)
```

```
         YHM                  DLSJ  TCSJ    YWXT     IP  \
0 "S1402048" 2014-11-04 08:44:46 NaN 1.225790e+17    221.205.98.55
1 S1411023 2014-11-04 08:45:06 NaN  1.225790e+17  183.184.226.205
2 S1402048 2014-11-04 08:46:39  NaN          NaN    221.205.98.55
3 20031509  2014-11-04 08:47:41  NaN         NaN    222.31.51.200
4 S1405010 2014-11-04 08:49:03   NaN   1.225790e+17  120.207.64.3

        REMARK
0  单点登录研究生系统成功！
1  单点登录研究生系统成功！
2     用户名或密码错误。
3   统一身份用户登录成功！
4  单点登录研究生系统成功！
```

注意： TXT 文本文件要保存成 UTF-8 格式才会不报错。但此例还有另外一个问题，即第一个数据带有双引号，该怎么处理？

(2) 导入 CSV 文件：

```
read_csv(file,names=[列名1,列名2,...],sep="",...)
```

file 为文件路径与文件名。

names 为列名，默认为文件中的第一行作为列名。

sep 为分隔符，默认为空，表示默认导入为一列。

【例 4-2】读取(导入)CSV 文件：

```
from pandas import read_csv
df = read_csv('E://rz20.csv',
names=['YHM','DLSJ','TCSJ','YWXT','IP','REMARK'],sep=",")
print(df)

   YHM   DLSJ TCSJ   YWXT  IP  REMARK
0  id   band  num  price NaN     NaN
1    1  130联通  123    159 NaN     NaN
2    2    131  124    753 NaN     NaN
3    3    132  125    456 NaN     NaN
4    4  133电信  126    852 NaN     NaN
```

使用 read_table 命令也能执行，结果与 read_csv 一致：

```
from pandas import read_table
```

```
df = read_table('E://rz20.csv',
names=['YHM','DLSJ','TCSJ','YWXT','IP','REMARK'],sep=",")
print(df)

   YHM   DLSJ TCSJ   YWXT  IP  REMARK
0  id   band  num  price NaN     NaN
1    1  130联通  123    159 NaN     NaN
2    2    131  124    753 NaN     NaN
3    3    132  125    456 NaN     NaN
4      4  133电信  126    852 NaN     NaN
```

(3) 导入 excel 文件：

```
read_excel(file, sheetname,header=0)
```

file 为文件路径与文件名。

sheetname 为 sheet 的名称，例如 sheet1。

header 为列名，默认为 0，文件的第一行作为列名。只接受布尔型 0 和 1。

【例 4-3】读取(导入)Excel 文件：

```
from pandas import read_excel
df = read_excel('E://rz1.xlsx',sheetname='Sheet2',header=1)
print(df)

   S1411023  2014-11-04 08:45:06  0122579031373493731 183.184.226.205  \
0  S1402048  2014-11-04 08:46:39                  NaN   221.205.98.55
1  20031509  2014-11-04 08:47:41                  NaN   222.31.51.200
2  S1405010  2014-11-04 08:49:03         1.225790e+17    120.207.64.3
3  20031509  2014-11-04 08:47:41                  NaN   222.31.51.200
4  S1405010  2014-11-04 08:49:03         1.225790e+17    120.207.64.3

   单点登录研究生系统成功!
0      用户名或密码错误。
1    统一身份用户登录成功!
2  单点登录研究生系统成功!
3    统一身份用户登录成功!
4  单点登录研究生系统成功!
```

注意： header 取 0 和 1 有差别，取 0 表示第一行作为表头显示，取 1 表示第一行丢弃，不作为表头显示。有时可以跳过首行或者读取多个表，例如：

```
df = pd.read_excel(filefullpath, sheetname=[0,2],skiprows=[0])
```

sheetname 可以指定读取几个 sheet，数目从 0 开始，如果 sheetname=[0,2]，则代表读取第 0 页和第 2 页的 sheet；skiprows=[0]代表读取时跳过第 0 行。

(4)　导入 MySQL 库：

```
read_sql(sql,con=数据库)
```

sql 为从数据库中查询数据的 SQL 语句。

con 为数据库的连接对象，需要在程序中选择创建。

示例代码如下：

```
import pandas
import MySQLdb
connection  = MySQLdb.connect(
    host = '127.0.0.1',              #本机的访问地址
    user = 'root',                   #登录名
    passwd = '',                     #访问密码，此处无密码
    db = 'test',                     #访问的数据库
    port = 5029,                     #访问端口
    charset = 'utf8')                #编码格式
data = pandas.read_sql("select * from t_user;",con = connection)
                                  # t_user 是 test 库中的表
connection.close()                #调用完要关闭数据库
```

4.2.4　数据导出

(1)　导出 CSV 文件：

```
to_csv(file_path,sep= ", ",index=TRUE,header=TRUE)
```

file_path 为文件路径。

sep 为分隔符，默认是逗号。

index 代表是否导出行序号，默认是 TRUE，导出行序号。

header 代表是否导出列名，默认是 TRUE，导出列名。

【例 4-4】导出 CSV 文件：

```
from pandas import DataFrame
from pandas import Series
df = DataFrame(
    {'age':Series([26,85,64]),'name':Series(['Ben','John','Jerry'])})
print(df)

   age   name
```

```
0   26    Ben
1   85   John
2   64  Jerry

df.to_csv('e:\\01.csv')                       #默认带上 index
df.to_csv('e:\\02.csv',index=False)           #无 index
```

结果如图 4-9 所示。

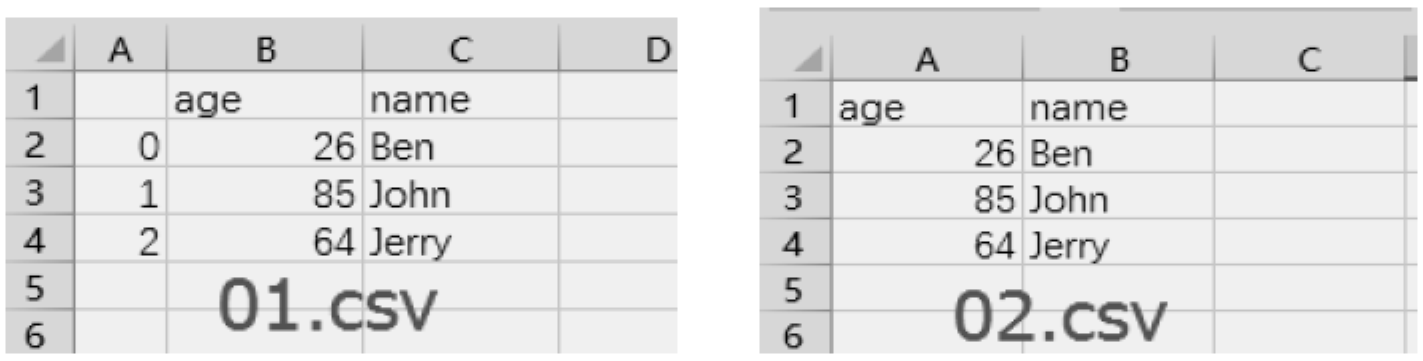


图 4-9　导出数据 01.csv 和 02.csv 的结果

(2)　导出 Excel 文件：

```
to_excel(file_path, index=TRUE,header=TRUE)
```

file_path 为文件路径。

index 表示是否导出行序号，默认是 TRUE，导出行序号。

header 表示是否导出列名，默认是 TRUE，导出列名。

【例 4-5】导出 Excel 文件：

```
from pandas import DataFrame
from pandas import Series
df = DataFrame(
    {'age':Series([26,85,64]),
    'name':Series(['Ben','John','Jerry'])})
df.to_excel('e:\\01.xlsx')                      #默认带上 index
df.to_excel('e:\\02.xlsx',index=False)          #无 index
```

结果如图 4-10 所示。

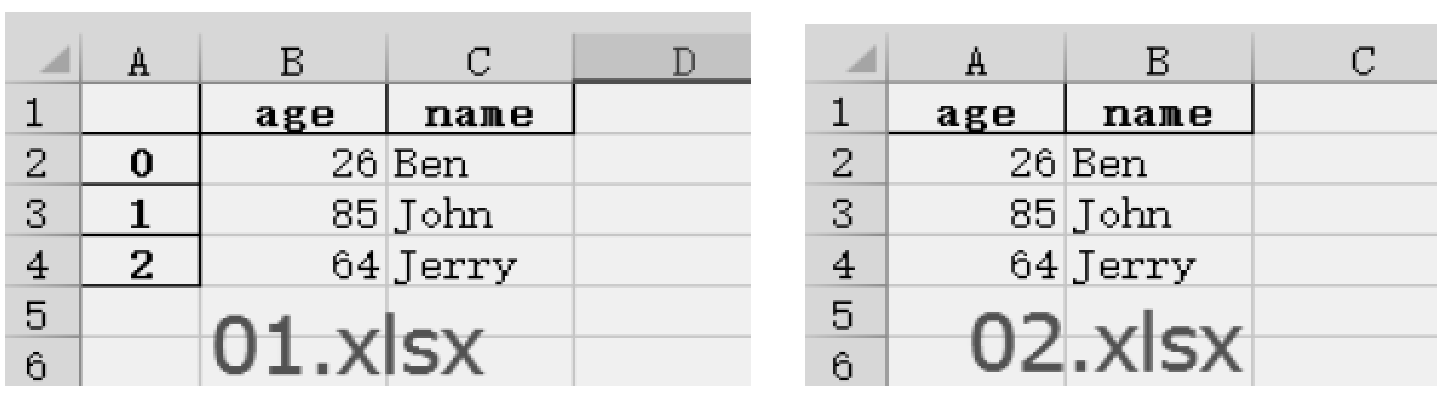


图 4-10　导出数据 01.xlsx 和 02.xlsx 结果图

注意： 凡是在 Python 2.7 中要写中文字符串的地方，都要在前面加 u。在 to_csv 里，就要多加 encoding = ‘UTF8’ 这个参数；若要用 Excel 直接打开，那么

encoding = “GBK”，或者 encoding = “GB2312”，因 Excel 默认的是这种编码。在 Python 3.4 后就不需要了。

(3)　导出到 MySQL 库：

```
to_sql(tableName, con=数据库链接)
```

tableName 为数据库中的表名。

con 表示数据库的连接对象，需要在程序中选择创建。

示例代码如下：

```
# -*- coding: utf-8 -*-
import MySQLdb
from pandas import DataFrame

connection = MySQLdb.connect(
    host='127.0.0.1',
    port=5029,
    user='root',
    passwd='',
    db='test',
    charset='utf8')

connection.autocommit(True) #自动递交数据连接
df = DataFrame({
    'age': [30, 22, 43],
    'name': ['Jhon', 'jerry', 'Ben']
    });
df.to_sql("table_1", connection, flavor='mysql', if_exists='append')
              #导入 MySQL 数据库 test 库下的 table_1 表中，以 append 追加的模式
connection.close()
```

4.3　数据处理

4.3.1　数据清洗

数据分析的第一步是提高数据质量。数据清洗就是处理缺失数据以及清除无意义的信息。这是数据价值链中最关键的步骤。垃圾数据，即使是通过最好的分析，也将产生错误的结果，并误导业务本身。因此在数据分析过程中，数据清洗占据了很大的工作量。

1．重复值的处理

drop_duplicates()：把数据结构中行相同的数据去除(保留其中的一行)。

【例 4-6】数据去重：

```
from pandas import DataFrame
from pandas import read_excel
df = read_excel('e://rz2.xlsx')
df
Out[1]:
        YHM         TCSJ         YWXT             IP  \
0   S1402048  18922254812  1.225790e+17   221.205.98.55
1   S1411023  13522255003  1.225790e+17   183.184.226.205
2   S1402048  13422259938          NaN    221.205.98.55
3   20031509  18822256753          NaN    222.31.51.200
4   S1405010  18922253721  1.225790e+17   120.207.64.3
5   20140007          NaN  1.225790e+17    222.31.51.200
6   S1404095  13822254373  1.225790e+17   222.31.59.220
7   S1402048  13322252452  1.225790e+17   221.205.98.55
8   S1405011  18922257681  1.225790e+17   183.184.230.38
9   S1402048  13322252452  1.225790e+17   221.205.98.55
10  S1405011  18922257681  1.225790e+17   183.184.230.38

newDF=df.drop_duplicates()
newDF
Out[2]:
        YHM         TCSJ         YWXT            IP
0  S1402048  18922254812  1.225790e+17    221.205.98.55
1  S1411023  13522255003  1.225790e+17    183.184.226.205
2  S1402048  13422259938          NaN     221.205.98.55
3  20031509  18822256753          NaN     222.31.51.200
4  S1405010  18922253721  1.225790e+17    120.207.64.3
5  20140007          NaN  1.225790e+17     222.31.51.200
6  S1404095  13822254373  1.225790e+17    222.31.59.220
7  S1402048  13322252452  1.225790e+17    221.205.98.55
8  S1405011  18922257681  1.225790e+17    183.184.230.38
```

在上面的 df 中，第 7 行和第 9 行数据相同，第 8 和第 10 行数据相同，去重后，第 7、9 和 8、10 各保留一行数据。

2．缺失值处理

对于缺失数据的处理方式有数据补齐、删除对应行、不处理等方法。

(1) dropna()：去除数据结构中值为空的数据行。

【例 4-7】删除数据为空所对应的行：

```
from pandas import DataFrame
from pandas import read_excel
df = read_excel('e://rz2.xlsx')
newDF=df.dropna()
newDF
Out[3]:
        YHM         TCSJ          YWXT               IP  \
0   S1402048  18922254812  1.225790e+17     221.205.98.55
1   S1411023  13522255003  1.225790e+17     183.184.226.205
4   S1405010  18922253721  1.225790e+17     120.207.64.3
6   S1404095  13822254373  1.225790e+17     222.31.59.220
7   S1402048  13322252452  1.225790e+17     221.205.98.55
8   S1405011  18922257681  1.225790e+17     183.184.230.38
9   S1402048  13322252452  1.225790e+17     221.205.98.55
10  S1405011  18922257681  1.225790e+17     183.184.230.38
```

例中的 2、3、5 行有空值 NaN，已经被删除。

(2)　df.fillna()：用其他数值替代 NaN。

有些时候，空数据直接删除会影响分析的结果，可以对数据进行填补。

【例 4-8】使用数值或者任意字符替代缺失值：

```
from pandas import DataFrame
from pandas import read_excel
df = read_excel('e://rz2.xlsx')
df.fillna('?')
Out[4]:
        YHM         TCSJ         YWXT               IP              DLSJ
0   S1402048  1.89223e+10  1.22579e+17     221.205.98.55    2014-11-04 08:44:46
1   S1411023  1.35223e+10  1.22579e+17     183.184.226.205  2014-11-04 08:45:06
2   S1402048  1.34223e+10            ?     221.205.98.55    2014-11-04 08:46:39
3   20031509  1.88223e+10            ?     222.31.51.200    2014-11-04 08:47:41
4   S1405010  1.89223e+10  1.22579e+17     120.207.64.3     2014-11-04 08:49:03
5   20140007            ?  1.22579e+17     222.31.51.200    2014-11-04 08:50:06
6   S1404095  1.38223e+10  1.22579e+17     222.31.59.220    2014-11-04 08:50:02
7   S1402048  1.33223e+10  1.22579e+17     221.205.98.55    2014-11-04 08:49:18
8   S1405011  1.89223e+10  1.22579e+17     183.184.230.38   2014-11-04 08:14:55
9   S1402048  1.33223e+10  1.22579e+17     221.205.98.55    2014-11-04 08:49:18
10  S1405011  1.89223e+10  1.22579e+17     183.184.230.38   2014-11-04 08:14:55
```

如 2、3、5 行有空，用?替代了缺失值。

(3)　df.fillna(method='pad')：用前一个数据值替代 NaN。

【例 4-9】用前一个数据值替代缺失值：

```
from pandas import DataFrame
from pandas import read_excel
df = read_excel('e://rz2.xlsx')
df.fillna(method='pad')
Out[5]:
        YHM        TCSJ         YWXT            IP            DLSJ
0   S1402048  18922254812  1.225790e+17   221.205.98.55    2014-11-04 08:44:46
1   S1411023  13522255003  1.225790e+17   183.184.226.205  2014-11-04 08:45:06
2   S1402048  13422259938  1.225790e+17   221.205.98.55    2014-11-04 08:46:39
3   20031509  18822256753  1.225790e+17   222.31.51.200    2014-11-04 08:47:41
4   S1405010  18922253721  1.225790e+17   120.207.64.3     2014-11-04 08:49:03
5   20140007  18922253721  1.225790e+17   222.31.51.200    2014-11-04 08:50:06
6   S1404095  13822254373  1.225790e+17   222.31.59.220    2014-11-04 08:50:02
7   S1402048  13322252452  1.225790e+17   221.205.98.55    2014-11-04 08:49:18
8   S1405011  18922257681  1.225790e+17   183.184.230.38   2014-11-04 08:14:55
9   S1402048  13322252452  1.225790e+17   221.205.98.55    2014-11-04 08:49:18
10  S1405011  18922257681  1.225790e+17   183.184.230.38   2014-11-04 08:14:55
```

(4) df.fillna(method='bfill')：用后一个数据值替代 NaN。

与 pad 相反，bfill 表示用后一个数据代替 NaN。可以用 limit 限制每列可以替代 NaN 的数目。

【例 4-10】用后一个数据值替代 NaN：

```
from pandas import DataFrame
from pandas import read_excel
df = read_excel('e://rz2.xlsx')
df.fillna(method='bfill')
Out[6]:
        YHM        TCSJ         YWXT            IP              DLSJ
0   S1402048  18922254812  1.225790e+17   221.205.98.55    2014-11-04 08:44:46
1   S1411023  13522255003  1.225790e+17  183.184.226.205   2014-11-04 08:45:06
2   S1402048  13422259938  1.225790e+17   221.205.98.55    2014-11-04 08:46:39
3   20031509  18822256753  1.225790e+17   222.31.51.200    2014-11-04 08:47:41
4   S1405010  18922253721  1.225790e+17    120.207.64.3    2014-11-04 08:49:03
5   20140007  13822254373  1.225790e+17   222.31.51.200    2014-11-04 08:50:06
6   S1404095  13822254373  1.225790e+17   222.31.59.220    2014-11-04 08:50:02
7   S1402048  13322252452  1.225790e+17   221.205.98.55    2014-11-04 08:49:18
8   S1405011  18922257681  1.225790e+17  183.184.230.38    2014-11-04 08:14:55
9   S1402048  13322252452  1.225790e+17   221.205.98.55    2014-11-04 08:49:18
10  S1405011  18922257681  1.225790e+17  183.184.230.38    2014-11-04 08:14:55
```

(5) df.fillna(df.mean())：用平均数或者其他描述性统计量来代替 NaN。

【例 4-11】使用均值来填补数据：

```
from pandas import DataFrame
from pandas import read_excel
df = read_excel('e://rz2_0.xlsx')
df
Out[7]:
   No  math  physical  Chinese
0   1    76        85       78
1   2    85        56      NaN
2   3    76        95       85
3   4   NaN        75       58
4   5    87        52       68

df.fillna(df.mean())
Out[8]:
   No  math  physical  Chinese
0   1    76        85    78.00
1   2    85        56    72.25
2   3    76        95    85.00
3   4    81        75    58.00
4   5    87        52    68.00
```

(6) df.fillna(df.mean()[math: physical])：选择列进行缺失值的处理。

【例 4-12】为某列使用该列的均值来填补数据：

```
from pandas import DataFrame
from pandas import read_excel
df = read_excel('e://rz2_0.xlsx')
df.fillna(df.mean()['math':'physical'])
Out[9]:
   No  math  physical  Chinese
0   1    76        85       78
1   2    85        56      NaN
2   3    76        95       85
3   4    81        75       58
4   5    87        52       68
```

(7) strip()：清除字符型数据左右(首尾)指定的字符，默认为空格，中间的不清除。

【例 4-13】删除字符串左右或首位指定的字符：

```
from pandas import DataFrame
from pandas import read_excel
df = read_excel('e://rz2.xlsx')
newDF=df['IP'].str.strip()  #因为 IP 是一个对象，所以先转为 str
newDF
```

```
Out[4]:
0       221.205.98.55
1     183.184.226.205
2       221.205.98.55
3       222.31.51.200
4        120.207.64.3
5       222.31.51.200
6       222.31.59.220
7       221.205.98.55
8      183.184.230.38
9       221.205.98.55
10     183.184.230.38
Name: IP, dtype: object
```

4.3.2 数据抽取

(1) **字段抽取**——抽出某列上指定位置的数据，做成新的列：

```
slice(start,stop)
```

start 为开始位置。

stop 为结束位置。

【**例 4-14**】从数据中抽出某列：

```
from pandas import DataFrame
from pandas import read_excel
df = read_excel('e://rz2.xlsx')
df['TCSJ']=df['TCSJ'].astype(str)     # astype()转化类型
df['TCSJ']
Out[1]:
0     18922254812
1     13522255003
2     13422259938
3     18822256753
4     18922253721
5             nan
6     13822254373
7     13322252452
8     18922257681
9     13322252452
10    18922257681
Name: TCSJ, dtype: object
```

```
bands = df['TCSJ'].str.slice(0,3)
bands
Out[2]:
0     189
1     135
2     134
3     188
4     189
5     nan
6     138
7     133
8     189
9     133
10    189
Name: TCSJ, dtype: object

areas= df['TCSJ'].str.slice(3,7);
areas
Out[3]:
0     2225
1     2225
2     2225
3     2225
4     2225
5
6     2225
7     2225
8     2225
9     2225
10    2225
Name: TCSJ, dtype: object

tell= df['TCSJ'].str.slice(7,11);
tell
Out[4]:
0     4812
1     5003
2     9938
3     6753
4     3721
5
6     4373
7     2452
```

```
8     7681
9     2452
10    7681
Name: TCSJ, dtype: object
```

(2) **字段拆分**——按指定的字符 sep，拆分已有的字符串：

```
split(sep,n,expand=False)
```

sep 是用于分隔字符串的分隔符。

n 为分割后新增的列数。

expand 代表是否展开为数据框，默认为 False。

返回值：expand 为 True，返回 DaraFrame；为 False 返回 Series。

【例 4-15】拆分字符串为指定的列数：

```
from pandas import DataFrame
from pandas import read_excel
df = read_excel('e://rz2.xlsx')
newDF=df['IP'].str.strip()  #IP 先转为 str，再删除首位空格
newDF=df['IP'].str.split('.',1,True)  #按第一个'.'分成两列，1 表示新增的列数
newDF
Out[1]:
         0           1
0       221     205.98.55
1       183   184.226.205
2       221     205.98.55
3       222     31.51.200
4       120      207.64.3
5       222     31.51.200
6       222     31.59.220
7       221     205.98.55
8       183    184.230.38
9       221     205.98.55
10      183    184.230.38

newDF.columns = ['IP1','IP2-4']  #给第一第二列增加列名称
newDF
Out[2]:
       IP1         IP2-4
0       221     205.98.55
1       183   184.226.205
2       221     205.98.55
3       222     31.51.200
```

```
4       120       207.64.3
5       222      31.51.200
6       222      31.59.220
7       221      205.98.55
8       183     184.230.38
9       221      205.98.55
10      183     184.230.38
```

(3)　**记录抽取**——是指根据一定的条件，对数据进行抽取。

```
dataframe[condition]
```

condition 为过滤条件。

返回值：DataFrame。

常用的 condition 类型如下。

比较运算：<、>、>=、<=、!=，如 df[df.comments>10000)]。

范围运算：between(left,right)，如 df[df.comments.between(1000,10000)]。

空置运算：pandas.isnull(column)，如 df[df.title.isnull()]。

字符匹配：str.contains(patten, na=False)，如 df[df.title.str.contains('电台', na=False)]。

逻辑运算：&(与)、|(或)、not(取反)，如：

```
df[(df.comments>=1000)&(df.comments<=10000)]
```

上面这一句跟 df[df.comments.between(1000,10000)]等价。

【例 4-16】按条件抽取数据：

```
import pandas
from pandas import read_excel
df = read_excel('e://rz2.xlsx')
df[df.TCSJ==13322252452]
Out[2]:
      YHM        TCSJ          YWXT            IP
7  S1402048  13322252452  1.225790e+17  221.205.98.55
9  S1402048  13322252452  1.225790e+17  221.205.98.55

df[df.TCSJ>13500000000]
Out[3]:
       YHM         TCSJ          YWXT              IP  \
0   S1402048  18922254812  1.225790e+17     221.205.98.55
1   S1411023  13522255003  1.225790e+17    183.184.226.205
3   20031509  18822256753           NaN    222.31.51.200
4   S1405010  18922253721  1.225790e+17      120.207.64.3
```

```
6   S1404095  13822254373  1.225790e+17     222.31.59.220
8   S1405011  18922257681  1.225790e+17    183.184.230.38
10  S1405011  18922257681  1.225790e+17    183.184.230.38

                   DLSJ
0   2014-11-04 08:44:46
1   2014-11-04 08:45:06
3   2014-11-04 08:47:41
4   2014-11-04 08:49:03
6   2014-11-04 08:50:02
8   2014-11-04 08:14:55
10  2014-11-04 08:14:55

df[df.TCSJ.between(13400000000,13999999999)]
Out[4]:
      YHM       TCSJ        YWXT              IP              DLSJ
1 S1411023 13522255003 1.225790e+17 183.184.226.205 2014-11-04 08:45:06
2 S1402048 13422259938          NaN   221.205.98.55   2014-11-04 08:46:39
6 S1404095 13822254373 1.225790e+17  222.31.59.220  2014-11-04 08:50:02

df[df.YWXT.isnull()]
Out[5]:
        YHM         TCSJ     YWXT            IP                DLSJ
2  S1402048  13422259938   NaN    221.205.98.55  2014-11-04 08:46:39
3  20031509  18822256753   NaN    222.31.51.200  2014-11-04 08:47:41

df[df.IP.str.contains('222.',na=False)]
Out[6]:
      YHM        TCSJ            YWXT             IP             DLSJ
3 20031509  18822256753            NaN   222.31.51.200   2014-11-04 08:47:41
5 20140007        NaN     1.225790e+17   222.31.51.200   2014-11-04 08:50:06
6 S1404095  13822254373   1.225790e+17  222.31.59.220   2014-11-04 08:50:02
```

(4)　**随机抽样**——是指随机从数据中按照一定的行数或者比例抽取数据：

```
numpy.random.randint(start,end,num)
```

start 为范围的开始值。

end 为范围的结束值。

num 为抽样个数。

返回值：行的索引值序列。

【例 4-17】随机抽取数据：

```
import numpy
import pandas
from pandas import read_excel
df = read_excel('e://rz2.xlsx')
df
Out[1]:
      YHM      TCSJ        YWXT            IP             DLSJ
0 S1402048 18922254812 1.225790e+17  221.205.98.55   2014-11-4 8:44
1 S1411023 13522255003 1.225790e+17 183.184.226.205 2014-11-4 8:45
2 S1402048 13422259938          NaN  221.205.98.55    2014-11-4 8:46
3 20031509 18822256753          NaN  222.31.51.200    2014-11-4 8:47
4 S1405010 18922253721 1.225790e+17  120.207.64.3    2014-11-4 8:49
5 20140007         NaN  1.225790e+17  222.31.51.200  2014-11-4 8:50
6 S1404095 13822254373 1.225790e+17  222.31.59.220  2014-11-4 8:50
7 S1402048 13322252452 1.225790e+17 221.205.98.55   2014-11-4 8:49
8 S1405011 18922257681 1.225790e+17 183.184.230.38  2014-11-4 8:14
9 S1402048 13322252452 1.225790e+17 221.205.98.55   2014-11-4 8:49
10S1405011 18922257681 1.225790e+17 183.184.230.38 2014-11-4 8:14

r = numpy.random.randint(0,10,3)
r
Out[2]: array([8, 2, 9])

df.loc[r,:]  #抽取 r 行数据
Out[3]:
     YHM        TCSJ        YWXT             IP              DLSJ
8 S1405011 18922257681 1.225790e+17 183.184.230.38 2014-11-04 08:14:55
2 S1402048 13422259938          NaN  221.205.98.55  2014-11-04 08:46:39
9 S1402048 13322252452 1.225790e+17 221.205.98.55  2014-11-04 08:49:18
```

如下我们来说明如何按照指定条件抽取数据。

① 使用 index 标签选取数据：df.loc[行标签,列标签]。例如：

```
df.loc['a':'b']          #选取 ab 两行数据，假设 a、b 为行索引
df.loc[:,'TCSJ']         #选取 TCSJ 列的数据
```

df.loc 的第一个参数是行标签，第二个参数为列标签(可选参数，默认为所有列标签)，两个参数既可以是列表，也可以是单个字符，如果两个参数都为列表，则返回的是 DataFrame，否则为 Series。

② 使用切片位置选取数据：df.iloc[行位置,列位置]。例如：

```
df.iloc[1,1]        #选取第二行，第二列的值，返回的为单个值
df.iloc[[0,2],:]  #选取第一行和第三行的数据
```

```
df.iloc[0:2,:]      #选取第一行到第三行(不包含)的数据
df.iloc[:,1]        #选取所有记录的第一列的值，返回的为一个 Series
df.iloc[1,:]        #选取第一行数据，返回的为一个 Series
```

说明： loc 为 location 的缩写，iloc 则为 integer & location 的缩写。更广义的切片方式是使用.ix，它自动根据给出的索引类型判断是使用位置还是标签进行切片。即 iloc 为整型索引；loc 为字符串索引；ix 是 iloc 和 loc 的合体。

Python 默认的行序号是从 0 开始的，我们称为行**位置**；但实际上 0 开始的行我们在计数时为第 1 行，也称为**行号**，是从 1 开始的；有时 index 是被命名的，如'one'、'two'、'three'、'four'或'a'、'b'、'c'、'd'等字符串，我们称其为**标签**。loc 索引的是行号、标签，不是行位置，如下例中 df2.loc[1]索引的是第一行(行号为 1)，其实位置为 0 行；iloc 索引的是位置，不能是标签或行号；ix 则三者皆可。

```
import pandas as pd
index_loc = ['a','b']
index_iloc = [1,2]
data = [[1,2,3,4],[5,6,7,8]]
columns = ['one','two','three','four']
df1 = pd.DataFrame(data=data,index=index_loc,columns=columns)
df2 = pd.DataFrame(data=data,index=index_iloc,columns=columns)

print(df1.loc['a'])
one      1
two      2
three    3
four     4
Name: a, dtype: int64

print(df1.iloc['a'])    #iloc 不能索引字符串
Traceback (most recent call last):
TypeError: cannot do label indexing on <class 'pandas.core.index.Index'>
with these indexers [a] of <class 'str'>

print(df2.iloc[1])      #iloc 索引的是行位置
one      5
two      6
three    7
four     8
Name: 2, dtype: int64

print(df2.loc[1])  #loc[1]索引的是行号，的对应的行位置为 0 行
```

```
one      1
two      2
three    3
four     4
Name: 1, dtype: int64

print(df1.ix[0])
one      1
two      2
three    3
four     4
Name: a, dtype: int64

print(df1.ix['a'])
one      1
two      2
three    3
four     4
Name: a, dtype: int64
```

③　通过逻辑指针进行数据切片：df[逻辑条件]。例如：

```
df[df.TCSJ >= 18822256753]   #单个逻辑条件
df[(df.TCSJ >=13422259938)&(df.TCSJ < 13822254373)]  #多个逻辑条件组合
```

这种方式获取的数据切片都是 DataFrame。例如：

```
df[df.TCSJ >= 18822256753]
Out[14]:
       YHM       TCSJ        YWXT           IP            DLSJ
0  S1402048 18922254812 1.225790e+17  221.205.98.55 2014-11-04 08:44:46
3  20031509 18822256753          NaN   222.31.51.200 2014-11-04 08:47:41
4  S1405010 18922253721 1.225790e+17  120.207.64.3 2014-11-04 08:49:03
8  S1405011 18922257681 1.225790e+17 183.184.230.38 2014-11-04 08:14:55
10 S1405011 18922257681 1.225790e+17 183.184.230.38 2014-11-04 08:14:55
```

(5)　**字典数据**——将字典数据抽取为 dataframe，有三种方法：

```
import pandas
from pandas import DataFrame

#1. 字典的 key 和 value 各作为一列
d1={'a':'[1,2,3]','b':'[0,1,2]'}
a1=pandas.DataFrame.from_dict(d1, orient='index')
  #将字典转化为 dataframe，且 key 列做成了 index
a1.index.name = 'key'  #将 index 的列名改成'key'
```

```
b1=a1.reset_index()     #重新增加 index，并将原 index 做成了'key'列
b1.columns=['key','value']   #对列重新命名为'key'和'value'
b1
Out[1]:
   key  value
0   b  [0, 1,2]
1   a  [1,2,3]

#2. 字典里的每一个元素作为一列(同长)
d2={'a':[1,2,3],'b':[4,5,6]}    #字典的 value 必须长度相等
a2= DataFrame(d2)
a2
Out[2]:
   a  b
0  1  4
1  2  5
2  3  6

#3. 字典里的每一个元素作为一列(不同长)
d = {'one' : pandas.Series([1,2,3]),'two' : pandas.Series([1,2,3,4])}
  #字典的 value 长度可以不相等
df = pandas.DataFrame(d)
df
Out[3]:
   one   two
0  1.0    1
1  2.0    2
2  3.0    3
3  NaN    4
```

也可以像下面这样处理：

```
import pandas
from pandas import Series
import numpy as np
from pandas import DataFrame

d = dict( A = np.array([1,2]), B = np.array([1,2,3,4]) )
DataFrame(dict([(k,Series(v)) for k,v in d.items()]))

Out[4]:
     A   B
0  1.0  1
1  2.0  2
```

```
2  NaN  3
3  NaN  4
```

还可以处理如下：

```
import numpy as np
import pandas as pd

my_dict = dict(A = np.array([1,2]), B = np.array([1,2,3,4]))
df = pd.DataFrame.from_dict(my_dict, orient='index').T
df
Out[5]:
     A    B
0  1.0  1.0
1  2.0  2.0
2  NaN  3.0
3  NaN  4.0
```

4.3.3 排名索引

1. 排名排序

Series 的 sort_index(ascending=True)方法可以对 index 进行排序操作，ascending 参数用于控制升序或降序，默认为升序。

在 DataFrame 上，.sort_index(axis=0, by=None, ascending=True)方法多了一个轴向的选择参数，以及一个 by 参数，by 参数的作用，是针对某一(些)列进行排序(不能对行使用 by 参数)。

例如：

```
from pandas import DataFrame
df0={'Ohio':[0,6,3],'Texas':[7,4,1],'California':[2,8,5]}
df=DataFrame(df0,index=['a','c','d'])
df
Out[1]:
   California  Ohio  Texas
a           2     0      7
c           8     6      4
d           5     3      1

df.sort_index(by='Ohio')

Out[2]:
```

```
   California  Ohio  Texas
a           2     0      7
d           5     3      1
c           8     6      4

df.sort_index(by=['California','Texas'])
Out[3]:
   California  Ohio  Texas
a           2     0      7
d           5     3      1
c           8     6      4

df.sort_index(axis=1)
Out[4]:
   California  Ohio  Texas
a           2     0      7
c           8     6      4
d           5     3      1
```

排名(Series.rank(method=‘average’, ascending=True))的作用与排序的不同之处在于，它会把对象的 values 替换成名次(从 1 到 n)，对于平级项，可以通过方法里的 method 参数来处理，method 参数有 4 个可选项：average、min、max、first。举例如下：

```
>>> ser=Series([3,2,0,3],index=list('abcd'))
>>> ser
a    3
b    2
c    0
d    3
dtype: int64
>>> ser.rank()
a    3.5
b    2.0
c    1.0
d    3.5
dtype: float64
>>> ser.rank(method='min')
a    3
b    2
c    1
d    3
dtype: float64
>>> ser.rank(method='max')
```

```
a    4
b    2
c    1
d    4
dtype: float64
>>> ser.rank(method='first')
a    3
b    2
c    1
d    4
dtype: float64
>>>
```

注意： 在 ser[0]和 ser[3]这对平级项上，不同 method 参数表现出的不同名次。DataFrame 的.rank(axis=0, method='average', ascending=True)方法多了 axis 参数，可选择按行或者按列分别进行排名，暂时好像没有针对全部元素的排名方法。

2．重新索引

Series 对象的重新索引通过其 .reindex(index=None,**kwargs)方法来实现。**kwargs 中常用的参数有两个：method=None 和 fill_value=np.NaN。

例如：

```
>>>from pandas import Series
>>>ser = Series([4.5,7.2,-5.3,3.6],index=['d','b','a','c'])
>>> A = ['a','b','c','d','e']
>>> ser.reindex(A)
a   -5.3
b    7.2
c    3.6
d    4.5
e    NaN
dtype: float64
>>> ser = ser.reindex(A,fill_value=0)
a   -5.3
b    7.2
c    3.6
d    4.5
e    0.0
dtype: float64
>>> ser.reindex(A,method='ffill')
```

```
a   -5.3
b    7.2
c    3.6
d    4.5
e    4.5
dtype: float64
>>> ser.reindex(A,fill_value=0,method='ffill')
a   -5.3
b    7.2
c    3.6
d    4.5
e    4.5
dtype: float64
>>>
```

.reindex()方法会返回一个新对象，其 index 严格遵循给出的参数，method:{‘backfill’, ‘bfill’, ‘pad’, ‘ffill’, None}参数用于指定插值(填充)方式，当没有给出时，默认用 fill_value 填充，值为 NaN(ffill = pad，bfill = back fill，分别指插值时向前还是向后取值)。

DataFrame 对象的重新索引方法.reindex(index=None,columns=None,**kwargs)仅比 Series 多了一个可选的 columns 参数，用于给列索引。用法与上例 Series 类似，只不过插值方法 method 参数只能应用于行，即轴 axis = 0。

例如：

```
>>> state = ['Texas','Utha','California']
>>> df.reindex(columns=state,method='ffill')
    Texas  Utha  California
a       1   NaN           2
c       4   NaN           5
d       7   NaN           8

[3 rows x 3 columns]
>>> df.reindex(index=['a','b','c','d'],columns=state,method='ffill')
   Texas  Utha  California
a       1   NaN           2
b       1   NaN           2
c       4   NaN           5
d       7   NaN           8

[4 rows x 3 columns]
>>>
```

可不可以通过 df.T.reindex(index,method='**').T 这样的方式来实现在列上的插值呢？

答案是肯定的。另外要注意，使用 reindex(index,method='**')的时候，index 必须是单调的，否则就会引发一个 ValueError: Must be monotonic for forward fill，比如上例中的最后一次调用，如果使用 index=['a','b','d','c']，就会报错。

4.3.4　数据合并

(1) **记录合并**——是指两个结构相同的数据框合并成一个数据框，也就是在一个数据框中追加另一个数据框的数据记录：

```
concat([dataFrame1, dataFrame2, ...])
```

dataFrame1 为数据框。

返回值：DataFrame。

【例 4-18】合并两个数据框，即合并记录：

```
import pandas
from pandas import DataFrame
from pandas import read_excel

df1 = read_excel('e://rz2.xlsx')
df1
Out[1]:
         YHM          TCSJ          YWXT              IP
0   S1402048  18922254812  1.225790e+17     221.205.98.55
1   S1411023  13522255003  1.225790e+17   183.184.226.205
2   S1402048  13422259938           NaN    221.205.98.55
3   20031509  18822256753           NaN    222.31.51.200
4   S1405010  18922253721  1.225790e+17     120.207.64.3
5   20140007  13422259313  1.225790e+17     222.31.51.200
6   S1404095  13822254373  1.225790e+17     222.31.59.220
7   S1402048  13322252452  1.225790e+17   221.205.98.55
8   S1405011  18922257681  1.225790e+17   183.184.230.38
9   S1402048  13322252452  1.225790e+17   221.205.98.55
10  S1405011  18922257681  1.225790e+17   183.184.230.38

df2 = read_excel('e://rz3.xlsx')
df2
Out[2]:
        YHM          TCSJ          YWXT              IP
0  S1402011  18603514812  1.225790e+17     221.205.98.55
1  S1411022  13103515003  1.225790e+17   183.184.226.205
2  S1402033  13203559930           NaN     221.205.98.55
```

```
df=pandas.concat([df1,df2])
df
Out[3]:
        YHM         TCSJ         YWXT             IP
0   S1402048  18922254812  1.225790e+17     221.205.98.55
1   S1411023  13522255003  1.225790e+17    183.184.226.205
2   S1402048  13422259938           NaN     221.205.98.55
3   20031509  18822256753           NaN     222.31.51.200
4   S1405010  18922253721  1.225790e+17     120.207.64.3
5   20140007  13422259313  1.225790e+17     222.31.51.200
6   S1404095  13822254373  1.225790e+17     222.31.59.220
7   S1402048  13322252452  1.225790e+17    221.205.98.55
8   S1405011  18922257681  1.225790e+17    183.184.230.38
9   S1402048  13322252452  1.225790e+17    221.205.98.55
10  S1405011  18922257681  1.225790e+17    183.184.230.38
0   S1402011  18603514812  1.225790e+17    221.205.98.55
1   S1411022  13103515003  1.225790e+17    183.184.226.205
2   S1402033  13203559930           NaN     221.205.98.55
```

两个文件的数据记录都合并到一起了，实现了数据记录的“叠加”或者记录顺延。

(2) **字段合并**——是指对同一个数据框中不同的列进行合并，形成新的列。

```
X = x1+x2+...
```

x1 为数据列 1。

x2 为数据列 2。

返回值：Series，合并后的系列。要求合并的系列长度一致。

【例 4-19】多个字段合并成一个新的字段：

```
import pandas
from pandas import DataFrame
from pandas import read_csv

df = read_csv('e://rz4.csv',sep=" ",names=['band','area','num'])
df
Out[1]:
   band  area   num
0   189  2225  4812
1   135  2225  5003
2   134  2225  9938
3   188  2225  6753
4   189  2225  3721
```

```
5   134  2225  9313
6   138  2225  4373
7   133  2225  2452
8   189  2225  7681

df = df.astype(str)
tel=df['band']+df['area']+df['num']
tel
Out[2]:
0    18922254812
1    13522255003
2    13422259938
3    18822256753
4    18922253721
5    13422259313
6    13822254373
7    13322252452
8    18922257681
dtype: object
```

(3)　**字段匹配**——是指不同结构的数据框(两个或两个以上的数据框)，按照一定的条件进行合并，即追加列：

```
merge(x,y,left_on,right_on)
```

x 是第一个数据框。

y 是第二个数据框。

left_on 是第一个数据框的用于匹配的列。

right_on 是第二个数据框的用于匹配的列。

返回值：DataFrame。

【例 4-20】按指定唯一字段匹配增加列：

```
import pandas
from pandas import DataFrame
from pandas import read_excel
df1 = read_excel('e://rz2.xlsx',sheetname='Sheet3')
df1
Out[1]:
   id  band  num
0   1   130  123
1   2   131  124
2   4   133  125
3   5   134  126
```

```
df2 = read_excel('e://rz2.xlsx',sheetname='Sheet4')
df2
Out[2]:
   id  band  area
0   1   130   351
1   2   131   352
2   3   132   353
3   4   133   354
4   5   134   355
5   5   135   356

pandas.merge(df1,df2,left_on='id',right_on='id')
Out[3]:
   id  band_x  num  band_y  area
0   1     130  123     130   351
1   2     131  124     131   352
2   4     133  125     133   354
3   5     134  126     134   355
4   5     134  126     135   356
```

这里只匹配了有相同序号的行，如 df1 中没有 id=3，在结果中也没有 id=3，但是在 df2 中有两个 id=5，在结果中也有两个 id=5，但是只匹配第一个 id=5。

4.3.5 数据计算

(1) **简单计算**——通过对各字段进行加、减、乘、除等四则算术运算，计算出的结果作为新的字段，如表 4-4 所示。

表 4-4 字段之间的运算结果作为新的字段

id	num	price
1	123	159
2	124	753
3	125	456
4	126	852

id	num	price	result
1	123	159	19557
2	124	753	93372
3	125	456	57000
4	126	852	107352

例如：

```
from pandas import read_csv
df = read_csv('e://rz2.csv',sep=',')
df
Out[1]:
   id  band  num  price
0   1   130  123    159
```

```
1   2   131  124    753
2   3   132  125    456
3   4   133  126    852

result=df.price*df.num
result
Out[2]:
0     19557
1     93372
2     57000
3    107352
dtype: int64

df['result']=result
df
Out[3]:
   id  band  num  price  result
0   1   130  123    159   19557
1   2   131  124    753   93372
2   3   132  125    456   57000
3   4   133  126    852  107352
```

(2) **数据标准化**——是指将数据按照比例缩放，使之落入特定的区间，一般使用 0~1 的区间来标准化：

```
X*=(x-min)/(max-min)
```

例如：

```
from pandas import read_csv

df = read_csv('e://rz2.csv',sep=',')
df
Out[1]:
   id  band  num  price
0   1   130  123    159
1   2   131  124    753
2   3   132  125    456
3   4   133  126    852

scale=(df.price-df.price.min())/(df.price.max()-df.price.min())
scale
Out[2]:
0    0.000000
```

```
1    0.857143
2    0.428571
3    1.000000
Name: price, dtype: float64
```

4.3.6 数据分组

数据分组是根据数据分析对象的特征，按照一定的数据指标，把数据划分为不同的区间来进行研究，以揭示其内在的联系和规律性。简单地说：就是新增一列，将原来的数据按照其性质归入新的类别中。数据分组的语法如下：

```
cut(series,bins,right=True,labels=NULL)
```

series 为需要分组的数据。

bins 为分组的依据数据。

right 为分组的时候右边是否闭合。

labels 为分组的自定义标签，可以不自定义。

例如，现有数据如表 4-5 所示，将数据进行分组。

表 4-5 数据分组

序号	品牌	数据	价格
1	130	123	159
2	131	124	753
3	132	125	456
4	133	126	852

→

序号	品牌	数据	价格	类别
1	130	123	159	500 以下
2	131	124	753	500 以上
3	132	125	456	500 以下
4	133	126	852	500 以上

```
import pandas
#from pandas import DataFrame
from pandas import read_csv

df = read_csv('e://rz2.csv',sep=',')
df
Out[1]:
   id  band  num  price
0   1   130  123    159
1   2   131  124    753
2   3   132  125    456
3   4   133  126    852

bins=[min(df.price)-1,500,max(df.price)+1]
labels=["500 以下","500 以上"]
```

```
pandas.cut(df.price,bins)
Out[2]:
0    (158, 500]
1    (500, 853]
2    (158, 500]
3    (500, 853]
Name: price, dtype: category
Categories (2, object): [(158, 500] < (500, 853]]

pandas.cut(df.price,bins,right=False)
Out[3]:
0    [158, 500)
1    [500, 853)
2    [158, 500)
3    [500, 853)
Name: price, dtype: category
Categories (2, object): [[158, 500) < [500, 853)]

pa=pandas.cut(df.price,bins,right=False,labels=labels)
pa
Out[5]:
0    500 以下
1    500 以上
2    500 以下
3    500 以上
Name: price, dtype: category
Categories (2, object): [500 以下 < 500 以上]

df['label']=pandas.cut(df.price,bins,right=False,labels=labels)
df
Out[6]:
   id  band  num  price  label
0   1   130  123    159  500 以下
1   2   131  124    753  500 以上
2   3   132  125    456  500 以下
3   4   133  126    852  500 以上
```

4.3.7　日期处理

(1)　**日期转换**——是指将字符型的日期格式转换为日期格式数据的过程：

```
to_datetime(dateString,format)
```

format 格式如下。

- %Y：年份。
- %m：月份。
- %d：日期。
- %H：小时。
- %M：分钟。
- %S：秒。

【例 4-21】使用 to_datetime(df.注册时间, format='%Y/%m/%d')转换：

```
from pandas import read_csv
from pandas import to_datetime
df = read_csv('e://rz3.csv',sep=',',encoding='utf8')
df
Out[1]:
   num  price  year  month       date
0  123    159  2016      1   2016/6/1
1  124    753  2016      2   2016/6/2
2  125    456  2016      3   2016/6/3
3  126    852  2016      4   2016/6/4
4  127    210  2016      5   2016/6/5
5  115    299  2016      6   2016/6/6
6  102    699  2016      7   2016/6/7
7  201    599  2016      8   2016/6/8
8  154    199  2016      9   2016/6/9
9  142    899  2016     10  2016/6/10

df_dt = to_datetime(df.date,format="%Y/%m/%d")
df_dt
Out[2]:
0   2016-06-01
1   2016-06-02
2   2016-06-03
3   2016-06-04
4   2016-06-05
5   2016-06-06
6   2016-06-07
7   2016-06-08
8   2016-06-09
9   2016-06-10
Name: date, dtype: datetime64[ns]
```

注意 CSV 的格式是否是 utf8 格式，否则会报错。另外，CSV 里 date 的格式是文本(字符串)格式。

(2) **日期格式化**——是指将日期型的数据按照给定的格式转化为字符型的数据：

```
apply(lambda x:处理逻辑)
```

处理逻辑即 datetime.strftime(x,format)。

【例 4-22】日期型数据转化为字符型数据：

```
#df_dt = to_datetime(df.注册时间, format='%Y/%m/%d');
#df_dt_str = df_dt.apply(df.注册时间, format='%Y/%m/%d')

from pandas import read_csv
from pandas import to_datetime
from datetime import datetime

df = read_csv('e://rz3.csv',sep=',',encoding='utf8')
df_dt = to_datetime(df.date,format="%Y/%m/%d")

df_dt_str=df_dt.apply(lambda x: datetime.strftime(x,"%Y/%m/%d"))
                                       #apply 见后注
df_dt_str
Out[1]:
0    2016/06/01
1    2016/06/02
2    2016/06/03
3    2016/06/04
4    2016/06/05
5    2016/06/06
6    2016/06/07
7    2016/06/08
8    2016/06/09
9    2016/06/10
Name: date, dtype: object
```

注意： 当希望将函数 f 应用到 DataFrame 对象的行或列时，可以使用.apply(f, axis=0, args=(), **kwds)方法，axis=0 表示按列运算，axis=1 表示按行运算。例如：

```
from pandas import DataFrame
df=DataFrame({'ohio':[1,3,6],'texas':[1,4,5],'california':[2,5,8]},index
=['a','c','d'])
df
Out[1]:
```

```
   california  ohio  texas
a           2     1      1
c           5     3      4
d           8     6      5

f = lambda x:x.max()-x.min()
df.apply(f)  #默认按列运算，同 df.apply(f,axis=0)
Out[2]:
california    6
ohio         5
texas         4
dtype: int64

df.apply(f,axis=1)  #按行运算
Out[3]:
a    1
c    2
d    3
dtype: int64
```

(3) **日期抽取**——是指从日期格式里面抽取出需要的部分属性：

```
Data_dt.dt.property
```

属性取值的相关含义如下。

second　　1~60 秒，从 1 开始到 60。

minute　　1~60 分，从 1 开始到 60。

hour　　1~24 小时，从 1 开始到 24。

day　　1~31 日，一个月中第几天，从 1 开始到 31。

month　　1~12 月，从 1 开始到 12。

year　　年份。

weekday　　1~7，一周中的第几天，从 1 开始，最大为 7。

【例 4-23】对日期进行抽取：

```
from pandas import read_csv;
from pandas import to_datetime;
df = read_csv('e://rz3.csv',sep=',',encoding='utf8')
df
Out[1]:
   num  price  year  month      date
0  123    159  2016      1  2016/6/1
1  124    753  2016      2  2016/6/2
```

```
2  125    456  2016      3   2016/6/3
3  126    852  2016      4   2016/6/4
4  127    210  2016      5   2016/6/5
5  115    299  2016      6   2016/6/6
6  102    699  2016      7   2016/6/7
7  201    599  2016      8   2016/6/8
8  154    199  2016      9   2016/6/9
9  142    899  2016     10  2016/6/10

df_dt =to_datetime(df.date,format='%Y/%m/%d')
df_dt
Out[2]:
0   2016-06-01
1   2016-06-02
2   2016-06-03
3   2016-06-04
4   2016-06-05
5   2016-06-06
6   2016-06-07
7   2016-06-08
8   2016-06-09
9   2016-06-10
Name: date, dtype: datetime64[ns]

df_dt.dt.year
Out[3]:
0    2016
1    2016
2    2016
3    2016
4    2016
5    2016
6    2016
7    2016
8    2016
9    2016
Name: date, dtype: int64

df_dt.dt.day
Out[4]:
0     1
1     2
2     3
```

```
3     4
4     5
5     6
6     7
7     8
8     9
9    10
Name: date, dtype: int64

df_dt.dt.month
df_dt.dt.weekday
df_dt.dt.second
df_dt.dt.hour
```

4.4 数 据 分 析

4.4.1 基本统计

基本统计分析：又叫描述性统计分析，一般统计某个变量的最小值、第一个四分位值、中值、第三个四分位值、以及最大值。

describe()：为描述性统计分析函数。

常用的统计函数如下。

size：计数(此函数不需要括号)。

sum()：求和。

mean()：平均值。

var()：方差。

std()：标准差。

【例 4-24】数据的基本统计：

```
from pandas import read_csv
df = read_csv('e://rz3.csv',sep=',',encoding='utf8')
df
Out[1]:
   num  price  year  month      date
0  123    159  2016      1  2016/6/1
1  124    753  2016      2  2016/6/2
2  125    456  2016      3  2016/6/3
3  126    852  2016      4  2016/6/4
4  127    210  2016      5  2016/6/5
```

```
5  115   299  2016     6   2016/6/6
6  102   699  2016     7   2016/6/7
7  201   599  2016     8   2016/6/8
8  154   199  2016     9   2016/6/9
9  142   899  2016    10  2016/6/10

df.num.describe()
Out[2]:
count     10.00000
mean     133.90000
std       27.39201
min      102.00000
25%      123.25000
50%      125.50000
75%      138.25000
max      201.00000
Name: num, dtype: float64

df.num.size   #注意：这里没有括号()
Out[3]: 10

df.num.max()
Out[4]: 201

df.num.min()
Out[5]: 102

df.num.sum()
Out[6]: 1339

df.num.mean()
Out[7]: 133.9

df.num.var()
Out[8]: 750.3222222222221

df.num.std()
Out[9]: 27.392010189510046
```

4.4.2　分组分析

分组分析：是指根据分组字段将分析对象划分成不同的部分，以对比分析各组之间差

异性的一种分析方法。

常用的统计指标：计数、求和、平均值。

常用形式：

```
df.groupby(by=['分类 1','分类 2',...])['被统计的列'].agg({列别名 1：统计函数 1，
列别名 2：统计函数 2， ...})
```

by——用于分组的列。

[ ]——用于统计的列。

.agg——统计别名，显示统计值的名称，统计函数用于统计数据。

size——计数。

sum——求和。

mean——均值。

【例 4-25】分组分析：

```
import numpy
from pandas import read_excel

df = read_excel('e:\\rz4.xlsx')
df
Out[1]:
            学号        班级    姓名  性别  英语  体育  军训  数分  高代  解几  计算机
0   2308024241  23080242  成龙   男   76  78   77  40   23  60    89
1   2308024244  23080242  周怡   女   66  91   75  47   47  44    82
2   2308024251  23080242  张波   男   85  81   75  45   45  60    80
3   2308024249  23080242  朱浩   男   65  50   80  72   62  71    82
4   2308024219  23080242  封印   女   73  88   92  61   47  46    83
5   2308024201  23080242  迟培   男   60  50   89  71   76  71    82
6   2308024347  23080243  李华   女   67  61   84  61   65  78    83
7   2308024307  23080243  陈田   男   76  79   86  69   40  69    82
8   2308024326  23080243  余皓   男   66  67   85  65   61  71    95
9   2308024320  23080243  李嘉   女   62  60   90  60   67  77    95
10  2308024342  23080243  李上初  男  76  90   84  60   66  60    82
11  2308024310  23080243  郭窦   女   79  67   84  64   64  79    85
12  2308024435  23080244  姜毅涛  男  77  71   87  61   73  76    82
13  2308024432  23080244  赵宇   男   74  74   88  68   70  71    85
14  2308024446  23080244  周路   女   76  80   77  61   74  80    85
15  2308024421  23080244  林建祥  男  72  72   81  63   90  75    85
16  2308024433  23080244  李大强  男  79  76   77  78   70  70    89
17  2308024428  23080244  李侧通  男  64  96   91  69   60  77    83
18  2308024402  23080244  王慧   女   73  74   93  70   71  75    88
19  2308024422  23080244  李晓亮  男  85  60   85  72   72  83    89
```

```
df.groupby(by=['班级','性别'])['军训'].agg({'总分':numpy.sum,'人数':
numpy.size,'平均值':numpy.mean,'方差':numpy.var,'标准差':numpy.std,'最高分':
numpy.max,'最低分':numpy.min})

Out[2]:
                   标准差          方差    最高分  人数  最低分  总分      平均值
 班级     性别
23080242 女    12.020815  144.500000   92   2    75   167  83.500000
         男     6.184658   38.250000   89   4    75   321  80.250000
23080243 女     3.464102   12.000000   90   3    84   258  86.000000
         男     1.000000    1.000000   86   3    84   255  85.000000
23080244 女    11.313708  128.000000   93   2    77   170  85.000000
         男     5.076088   25.766667   91   6    77   509  84.833333
```

4.4.3　分布分析

分布分析是指根据分析的目的，将数据(定量数据)进行等距或不等距的分组，是研究各组分布规律的一种分析方法。

【例 4-26】分布分析：

```
import numpy
import pandas
from pandas import read_excel

df = read_excel('e:\\rz4.xlsx')
df
Out[1]:
        学号         班级       姓名  性别  英语  体育  军训  数分  高代  解几  计算机  总分
0   2308024241  23080242   成龙   男   76   78   77   40   23   60   89   443
1   2308024244  23080242   周怡   女   66   91   75   47   47   44   82   452
2   2308024251  23080242   张波   男   85   81   75   45   45   60   80   471
3   2308024249  23080242   朱浩   男   65   50   80   72   62   71   82   482
4   2308024219  23080242   封印   女   73   88   92   61   47   46   83   490
5   2308024201  23080242   迟培   男   60   50   89   71   76   71   82   499
6   2308024347  23080243   李华   女   67   61   84   61   65   78   83   499
7   2308024307  23080243   陈田   男   76   79   86   69   40   69   82   501
8   2308024326  23080243   余皓   男   66   67   85   65   61   71   95   510
9   2308024320  23080243   李嘉   女   62   60   90   60   67   77   95   511
10  2308024342  23080243  李上初  男   76   90   84   60   66   60   82   518
11  2308024310  23080243   郭窦   女   79   67   84   64   64   79   85   522
12  2308024435  23080244  姜毅涛  男   77   71   87   61   73   76   82   527
```

```
13  2308024432  23080244   赵宇  男  74   74  88   68   70   71   85   530
14  2308024446  23080244   周路  女  76   80  77   61   74   80   85   533
15  2308024421  23080244  林建祥  男  72   72  81   63   90   75   85   538
16  2308024433  23080244  李大强  男  79   76  77   78   70   70   89   539
17  2308024428  23080244  李侧通  男  64   96  91   69   60   77   83   540
18  2308024402  23080244   王慧  女  73   74  93   70   71   75   88   544
19  2308024422  23080244  李晓亮  男  85   60  85   72   72   83   89   546

bins = [min(df.总分)-1,450,500,max(df.总分)+1]   #将数据分成三段

bins
Out[3]: [442, 450, 500, 547]

labels=['450 及其以下','450 到 500','500 及其以上']  #给三段数据贴标签

labels
Out[5]: ['450 及其以下', '450 到 500', '500 及其以上']

总分分层 = pandas.cut(df.总分,bins,labels=labels)

总分分层
Out[7]:
0      450 及其以下
1      450 到 500
2      450 到 500
3      450 到 500
4      450 到 500
5      450 到 500
6      450 到 500
7      500 及其以上
8      500 及其以上
9      500 及其以上
10     500 及其以上
11     500 及其以上
12     500 及其以上
13     500 及其以上
14     500 及其以上
15     500 及其以上
16     500 及其以上
17     500 及其以上
18     500 及其以上
19     500 及其以上
Name: 总分, dtype: category
```

```
Categories (3, object): [450 及其以下 < 450 到 500 < 500 及其以上]

df['总分分层']= 总分分层
df
[Out8]:
          学号        班级    姓名 性别 英语 体育 军训 数分 高代 解几 计算机基础 总分  总分分层
   0  2308024241  23080242  成龙  男  76  78  77  40  23  60      89     443  450 及其以下
   1  2308024244  23080242  周怡  女  66  91  75  47  47  44      82     452  450 到 500
   2  2308024251  23080242  张波  男  85  81  75  45  45  60      80     471  450 到 500
   3  2308024249  23080242  朱浩  男  65  50  80  72  62  71      82     482  450 到 500
   4  2308024219  23080242  封印  女  73  88  92  61  47  46      83     490  450 到 500
   5  2308024201  23080242  迟培  男  60  50  89  71  76  71      82     499  450 到 500
   6  2308024347  23080243  李华  女  67  61  84  61  65  78      83     499  450 到 500
   7  2308024307  23080243  陈田  男  76  79  86  69  40  69      82     501  500 及其以上
   8  2308024326  23080243  余皓  男  66  67  85  65  61  71      95     510  500 及其以上
   9  2308024320  23080243  李嘉  女  62  60  90  60  67  77      95     511  500 及其以上
   10 2308024342  23080243  李上初 男  76  90  84  60  66  60      82     518  500 及其以上
   11 2308024310  23080243  郭窦  女  79  67  84  64  64  79      85     522  500 及其以上
   12 2308024435  23080244  姜毅涛 男  77  71  87  61  73  76      82     527  500 及其以上
   13 2308024432  23080244  赵宇  男  74  74  88  68  70  71      85     530  500 及其以上
   14 2308024446  23080244  周路  女  76  80  77  61  74  80      85     533  500 及其以上
   15 2308024421  23080244  林建祥 男  72  72  81  63  90  75      85     538  500 及其以上
   16 2308024433  23080244  李大强 男  79  76  77  78  70  70      89     539  500 及其以上
   17 2308024428  23080244  李侧通 男  64  96  91  69  60  77      83     540  500 及其以上
   18 2308024402  23080244  王慧  女  73  74  93  70  71  75      88     544  500 及其以上
   19 2308024422  23080244  李晓亮 男  85  60  85  72  72  83      89     546  500 及其以上

df.groupby(by=['总分分层'])['总分'].agg({'人数':numpy.size})
Out[9]:
             人数
总分分层
450 及其以下    1
450 到 500     6
500 及其以上   13
```

4.4.4 交叉分析

交叉分析通常用于分析两个或两个以上分组变量之间的关系，以交叉表形式进行变量间关系的对比分析。一般分为定量、定量分组交叉；定量、定性分组交叉；定性、定型分组交叉。交叉分析所使用的分析函数如下：

```
pivot_table(values,index,columns,aggfunc,fill_value)
```

values——数据透视表中的值。

index——数据透视表中的行。

columns——数据透视表中的列。

aggfunc——统计函数。

fill_value——NA 值的统一替换。

返回值：数据透视表的结果。

【例 4-27】交叉分析：

```
import numpy
import pandas
from pandas import read_excel
from pandas import pivot_table     #在 Spyder 下也可以不导入

df = read_excel('e:\\rz4.xlsx')
bins = [min(df.总分)-1,450,500,max(df.总分)+1]
labels=['450 及其以下','450 到 500','500 及其以上']
总分分层 = pandas.cut(df.总分,bins,labels=labels)
df['总分分层']= 总分分层
df.pivot_table(values=['总分'],index=['总分分层'],columns=['性别'],
aggfunc=[numpy.size,numpy.mean])
Out[1]:
            size              mean
                         总分            总分
性别         女    男         女              男
总分分层
450 及其以下  NaN  1         NaN          443.000000
450 到 500    3    3         480.333333   484.000000
500 及其以上  4    9         527.500000   527.666667

df.pivot_table(values=['总分'],index=['总分分层'],columns=['性别'],
aggfunc=[numpy.size,numpy.mean], fill_value=0)
#也可以将统计为 0 的赋值为零，默认为 nan
Out[2]:
             size            mean
                            总分            总分
性别          女   男          女              男
总分分层
450 及其以下   0   1        0.000000      443.000000
450 到 500     3   3        480.333333   484.000000
500 及其以上   4   9        527.500000    527.666667
```

4.4.5　结构分析

结构分析是在分组的基础上，计算各组成部分所占的比重，进而分析总体的内部特征的一种分析方法。

所使用的函数如下：

```
df_pt.sum(axis)
df_pt.div(df_pt.sum(axis),axis)
```

axis 参数说明：0 表示列；1 表示行。

【**例 4-28**】结构分析：

```
#假设要计算班级团体总分情况
import numpy
import pandas

from pandas import read_excel
from pandas import pivot_table  #在 Spyder 下也可以不导入

df = read_excel('e:\\rz4.xlsx')

df_pt = df.pivot_table(values=['总分'],index=['班级'],columns=['性别'],
aggfunc=[numpy.sum])

df_pt
Out[1]:
              sum
              总分
性别           女      男
班级
23080242   942  1895
23080243  1532  1529
23080244  1077  3220

df_pt.sum()
Out[2]:
           性别
sum  总分   女     3551
           男     6644
dtype: int64
```

```
df_pt.sum(axis=0) #效果同省略
Out[3]:
          性别
sum  总分  女     3551
          男     6644
dtype: int64

df_pt.sum(axis=1)
Out[4]:
班级
23080242    2837
23080243    3061
23080244    4297
dtype: int64

df_pt.div(df_pt.sum(axis=1),axis=0)  #按列占比
Out[5]:
                sum
                总分
性别         女          男
班级
23080242  0.332041  0.667959
23080243  0.500490  0.499510
23080244  0.250640  0.749360

df_pt.div(df_pt.sum(axis=0),axis=1)  #按行占比
Out[6]:
                sum
                总分
性别         女          男
班级
23080242  0.265277  0.285220
23080243  0.431428  0.230132
23080244  0.303295  0.484648
```

4.4.6 相关分析

相关分析是研究现象之间是否存在某种依存关系，并对具体有依存关系的现象探讨其相关方向以及相关程度，是研究随机变量之间相关关系的一种统计方法。

相关系数可以用来描述定量变量之间的关系。

相关系数与相关程度如表 4-6 所示。

表 4-6　相关系数与相关程度

相关系数|r|取值范围	相关程度
0≤|r|<0.3	低度相关
0.3≤|r|<0.8	中度相关
0.8≤|r|≤1	高度相关

相关分析函数：

```
DataFrame.corr()
Series.corr(other)
```

如果由数据框调用 corr 方法，那么将会计算每列两两之间的相似度。如果由序列调用 corr 方法，那么只是计算该序列与传入的序列之间的相关度。

返回值：

- DataFrame 调用将返回 DataFrame。
- Series 调用将返回一个数值型，大小为相关度。

【例 4-29】相关分析：

```
import numpy
import pandas
from pandas import read_excel
df = read_excel('e:\\rz4.xlsx')
df
Out[1]:
            学号        班级    姓名  性别  英语  体育  军训  数分  高代  解几  计算机基础  总分
0   2308024241  23080242   成龙   男   76   78   77   40   23   60     89   443
1   2308024244  23080242   周怡   女   66   91   75   47   47   44     82   452
2   2308024251  23080242   张波   男   85   81   75   45   45   60     80   471
3   2308024249  23080242   朱浩   男   65   50   80   72   62   71     82   482
4   2308024219  23080242   封印   女   73   88   92   61   47   46     83   490
5   2308024201  23080242   迟培   男   60   50   89   71   76   71     82   499
6   2308024347  23080243   李华   女   67   61   84   61   65   78     83   499
7   2308024307  23080243   陈田   男   76   79   86   69   40   69     82   501
8   2308024326  23080243   余皓   男   66   67   85   65   61   71     95   510
9   2308024320  23080243   李嘉   女   62   60   90   60   67   77     95   511
10  2308024342  23080243  李上初   男   76   90   84   60   66   60     82   518
11  2308024310  23080243   郭窦   女   79   67   84   64   64   79     85   522
12  2308024435  23080244  姜毅涛   男   77   71   87   61   73   76     82   527
13  2308024432  23080244   赵宇   男   74   74   88   68   70   71     85   530
14  2308024446  23080244   周路   女   76   80   77   61   74   80     85   533
15  2308024421  23080244  林建祥   男   72   72   81   63   90   75     85   538
```

```
16  2308024433  23080244  李大强  男  79  76  77  78  70  70    89  539
17  2308024428  23080244  李侧通  男  64  96  91  69  60  77    83  540
18  2308024402  23080244   王慧  女  73  74  93  70  71  75    88  544
19  2308024422  23080244  李晓亮  男  85  60  85  72  72  83    89  546

#两列之间的相关度计算
df['高代'].corr(df['数分'])
Out[2]: 0.60774082332601076

#多列之间的相关度计算
df.loc[:,['英语','体育','军训','计算机基础','解几','数分','高代']].corr()
Out[3]:
          英语        体育       军训     计算机基础      解几     数分       高代
英语   1.000000  0.244323 -0.335015 -0.119039  0.027452 -0.129588 -0.125245
体育   0.244323  1.000000 -0.111315 -0.266896 -0.526276 -0.369766 -0.382447
军训-0.335015 -0.111315  1.000000  0.148933  0.249299  0.469226  0.251903
计算机基础 -0.119039 -0.266896  0.148933  1.000000  0.305934  0.123399  0.096979
解几 0.027452 -0.526276  0.249299  0.305934  1.000000  0.544394  0.613281
数分-0.129588 -0.369766  0.469226  0.123399  0.544394  1.000000  0.607741
高代-0.125245 -0.382447  0.251903  0.096979  0.613281  0.607741  1.000000
```

4.5 数据可视化

4.5.1 饼图

饼图(Pie Graph)又称圆形图，是一个划分为几个扇形的圆形统计图，它能够直观地反映个体与总体的比例关系。绘制饼图的方法如下：

```
pie(x,labels,colors,explode,autopct)
```

x——进行绘图的序列。

labels——饼图的各部分标签。

colors——饼图的各部分颜色，使用 GRB 标颜色。

explode——需要突出的块状序列。

autopct——饼图占比的显示格式。例如%.2f：保留两位小数。

【例 4-30】绘制饼图：

```
import numpy
import matplotlib
import matplotlib.pyplot as plt
from pandas import read_csv
```

```
df = read_csv('e:\\rz20.csv',sep=',')
df
Out[1]:
   id   band     num   price
0    1  130 联通  123    159
1    2    131     124    753
2    3    132     125    456
3    4  133 电信  126    852

gb=df.groupby(by=['band'],as_index=False)['num'].agg({'price':numpy.size})
gb
Out[2]:
    band      price
0  130 联通       1
1    131         1
2    132         1
3  133 电信      1
#为了便于图中显示中文
font = {'family':'SimHei'}
matplotlib.rc('font',**font)
#画饼图
plt.pie(gb['price'],labels=gb['band'],autopct='%.2f%%')
plt.show()
```

结果如图 4-11 所示。

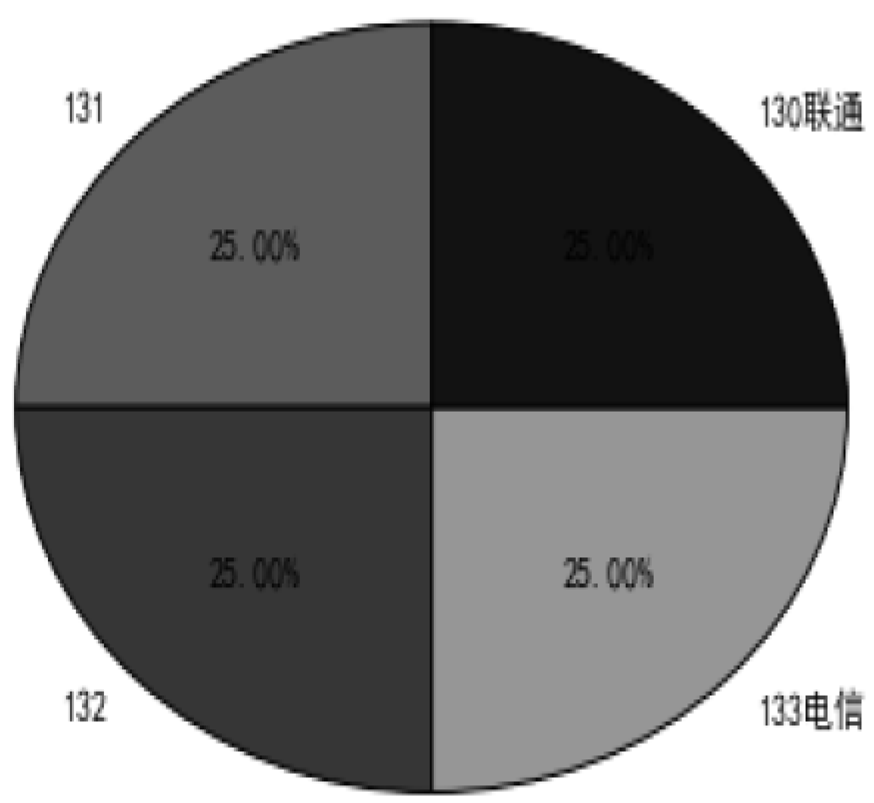


图 4-11　饼图

注意： 在画图时，所有的字段中的数据列含有中文的要注意它所保存的格式必须是 utf-8，否则会报错。可以用记事本打开看看的格式，方法如图 4-12 所示。

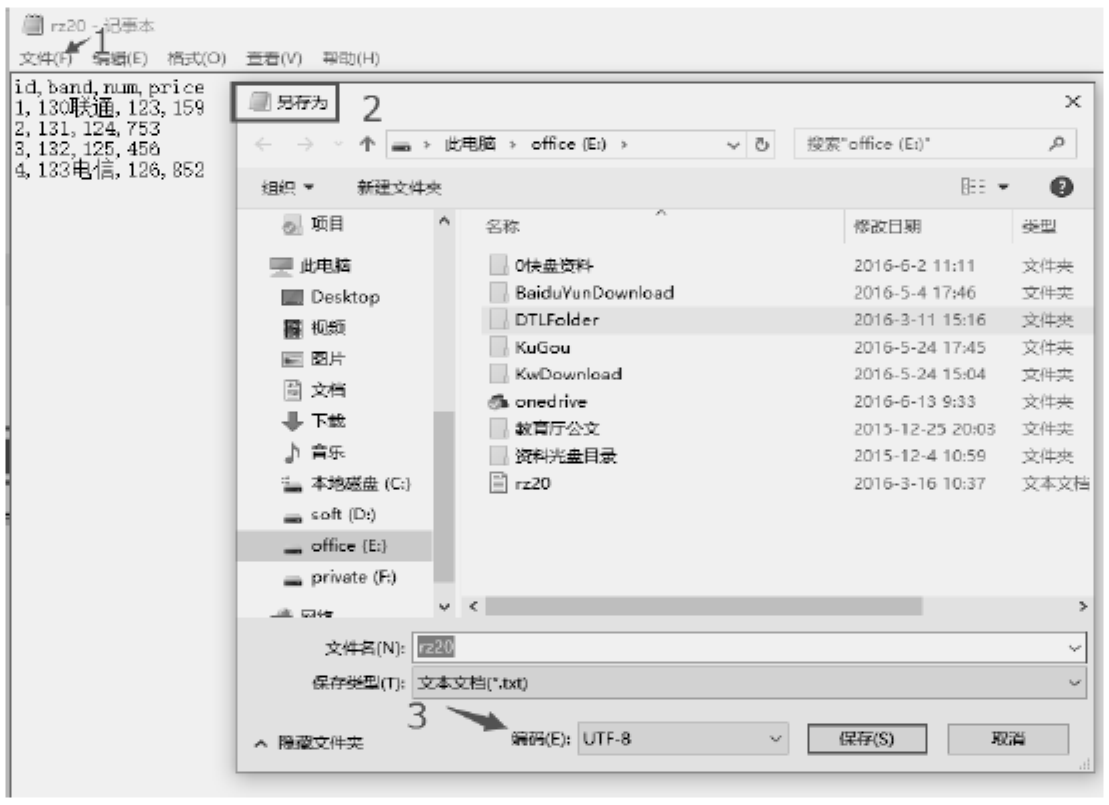


图 4-12　存储为 utf-8 格式

4.5.2　散点图

散点图(scatter diagram)是以一个变量为横坐标，另一个变量为纵坐标，利用散点(坐标点)的分布形态反映变量关系的一种图形。相关的方法如下：

```
plt.plot(x,y, '. ',color=(r,g,b))
plt.xlabel('x 轴坐标')
plt.ylabel('y 轴坐标')
plt.grid(True)
```

x、y——X 轴和 Y 轴的序列。

‘.’、‘o’——小点还是大点。

Color——散点图的颜色，可以用 RGB 定义，也可以用英文字母定义。

RGB 颜色的设置：(red,green,blue)，由红绿蓝颜色组成。

常用 GRB 颜色见表 4-7。

表 4-7　常用 GRB 颜色对照

颜色	英文	RGB	十六进制
白色	write	(1, 1, 1)	#FFFFFF
黑色	black	(0, 0, 0)	#000000
红色	red	(1, 0, 0)	#FF0000
橙色	orange	(1, 0.5, 0)	#FFA500
黄色	yellow	(1, 1, 0)	#FFFF00
绿色	green	(0, 1, 0)	#00FF00
蓝色	blue	(0, 0, 1)	#0000FF
靛色	indigo	(0.3, 0, 0.5)	#4B0082
紫色	purple	(0.63, 0.13, 0.95)	#A020F0

例如：

```
import matplotlib
import matplotlib.pyplot as plt
from pandas import read_csv

df = read_csv('e:\\rz20.csv',sep=',')

df
Out[1]:
    id  band  num  price
0   1  130联通  123   159
1   2    131  124   753
2   3    132  125   456
3   4  133电信  126   852

#为了便于图中显示中文
font = {'family':'SimHei'}
matplotlib.rc('font',**font)

#画图
plt.plot(df['price'],df['num'],'*')
plt.xlabel('price')
plt.ylabel('num')
plt.grid(True)
plt.show()
```

结果如图 4-13 所示。

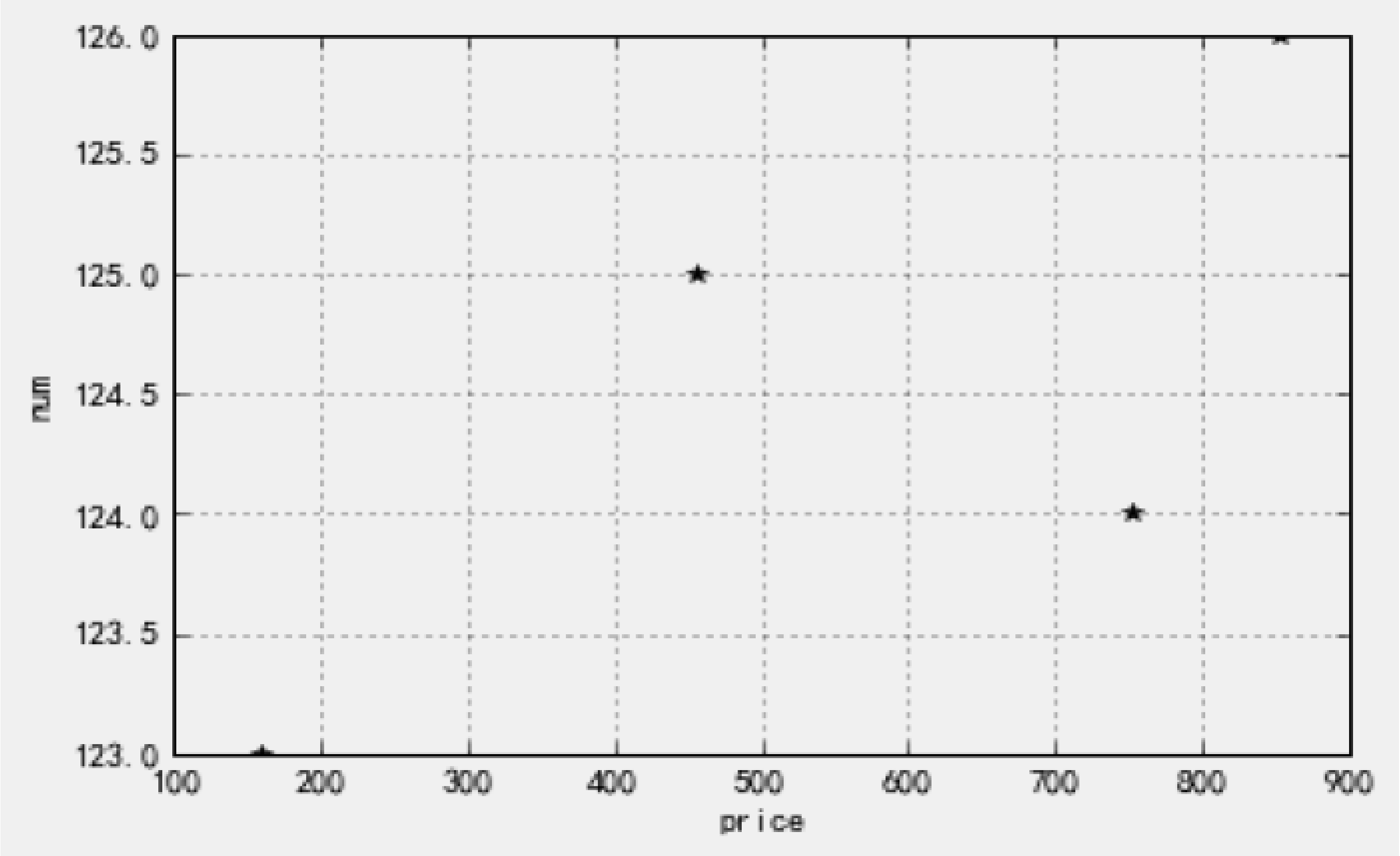


图 4-13　散点图

4.5.3 折线图

折线图也称趋势图，它是用直线段将各数据点连接起来而组成的图形，以折线方式显示数据的变化趋势。相关的方法如下：

```
plot(x,y, '-',color)
title('图的标题')
```

‘-’为画线的样式。有多种样式，详见表 4-8。

表 4-8 plot 函数画线样式释义

参 数 值	说 明
-	连续的曲线
--	连续的虚线
-.	连续的用带点的曲线
:	由点连成的曲线
.	小点，散点图
o	大点，散点图
,	像素点(更小的点)的散点图
*	五角星的点，散点图
>	右角标记散点图
<	左角标记散点图
1(2,3,4)	伞形上(下左右)标记散点图
s	正方形标记散点图
p	五角星标记散点图
v	下三角标记散点图
^	上三角标记散点图
h	多边形标记散点图
d	钻石标记散点图

下例主要是实现以学号的后三位为横轴，总分为纵轴，画折线图。分三步。

第一，实现提取学号后三位。

第二，为了实现按学号后三位排序，就得实现学号后三位与相应的总分构成一对，再排序，否则学号后三位排序了，但对应不上相应的总分。

第三，按照学号后三位与总分的序对顺序拆分成 list1 和 list2 两列，再把 list1 做成横轴，list2 做成纵轴。

【**例 4-31**】绘制折线图：

```
import matplotlib
from pandas import read_excel
from matplotlib import pyplot as plt

df = read_excel('e:\\rz4.xlsx',sep=',')
df
Out[1]:
       学号          班级       姓名  性别  英语  体育  军训  数分  高代  解几  计算机基础  总分
0   2308024241  23080242  成龙  男  76  78  77  40  23  60      89  443
1   2308024244  23080242  周怡  女  66  91  75  47  47  44      82  452
2   2308024251  23080242  张波  男  85  81  75  45  45  60      80  471
3   2308024249  23080242  朱浩  男  65  50  80  72  62  71      82  482
4   2308024219  23080242  封印  女  73  88  92  61  47  46      83  490
5   2308024201  23080242  迟培  男  60  50  89  71  76  71      82  499
6   2308024347  23080243  李华  女  67  61  84  61  65  78      83  499
7   2308024307  23080243  陈田  男  76  79  86  69  40  69      82  501
8   2308024326  23080243  余皓  男  66  67  85  65  61  71      95  510
9   2308024320  23080243  李嘉  女  62  60  90  60  67  77      95  511
10  2308024342  23080243  李上初  男  76  90  84  60  66  60      82  518
11  2308024310  23080243  郭窦  女  79  67  84  64  64  79      85  522
12  2308024435  23080244  姜毅涛  男  77  71  87  61  73  76      82  527
13  2308024432  23080244  赵宇  男  74  74  88  68  70  71      85  530
14  2308024446  23080244  周路  女  76  80  77  61  74  80      85  533
15  2308024421  23080244  林建祥  男  72  72  81  63  90  75      85  538
16  2308024433  23080244  李大强  男  79  76  77  78  70  70      89  539
17  2308024428  23080244  李侧通  男  64  96  91  69  60  77      83  540
18  2308024402  23080244  王慧  女  73  74  93  70  71  75      88  544
19  2308024422  23080244  李晓亮  男  85  60  85  72  72  83      89  546

#提取学号后三位并打印出来
def right3(df,a):
    '''
    实现提取学号后三位
    '''
    list0=[]
    list1=list(df[a])
    for i in list1:
        i=str(i)
        list2=i[-3:]
        list0.append(list2)
    return list0
```

```
df_ri=right3(df,'学号')
df_ri
Out[2]:
['241',
 '244',
 '251',
 '249',
 '219',
 '201',
 '347',
 '307',
 '326',
 '320',
 '342',
 '310',
 '435',
 '432',
 '446',
 '421',
 '433',
 '428',
 '402',
 '422']

#提取“总分”并转化为列表
df_va=list(df['总分'])
df_va
Out[3]:
[443,
 452,
 471,
 482,
 490,
 499,
 499,
 501,
 510,
 511,
 518,
 522,
 527,
 530,
```

```
 533,
 538,
 539,
 540,
 544,
 546]

#构成“学号后三位”与“总分”对应的 list
def dic3(a,b):
    '''
    组成数对，并排序
    '''
    t=[]
    for k in range(len(a)):
        d=(a[k],b[k])
        t.append(d)
    t.sort()
    return t
df_di=dic3(df_ri,df_va)
df_di
Out[4]:
[('201', '499'),
 ('219', '490'),
 ('241', '443'),
 ('244', '452'),
 ('249', '482'),
 ('251', '471'),
 ('307', '501'),
 ('310', '522'),
 ('320', '511'),
 ('326', '510'),
 ('342', '518'),
 ('347', '499'),
 ('402', '544'),
 ('421', '538'),
 ('422', '546'),
 ('428', '540'),
 ('432', '530'),
 ('433', '539'),
 ('435', '527'),
 ('446', '533')]

#将数对拆成两列
```

```
list1=[df_di[i][0] for i in range(len(df_di))]
list2=[df_di[i][1] for i in range(len(df_di))]

#为了便于图中显示中文
font = {'family':'SimHei'}
matplotlib.rc('font',**font)

#用'-'画顺滑的曲线，list1 作为横轴，list2 作为纵轴
plt.plot(list1,list2,'-')
plt.title('学号与总分折线图')  #图的标题
plt.show()

Out[5]: [<matplotlib.lines.Line2D at 0x2c7e463e780>]
```

结果如图 4-14 所示。

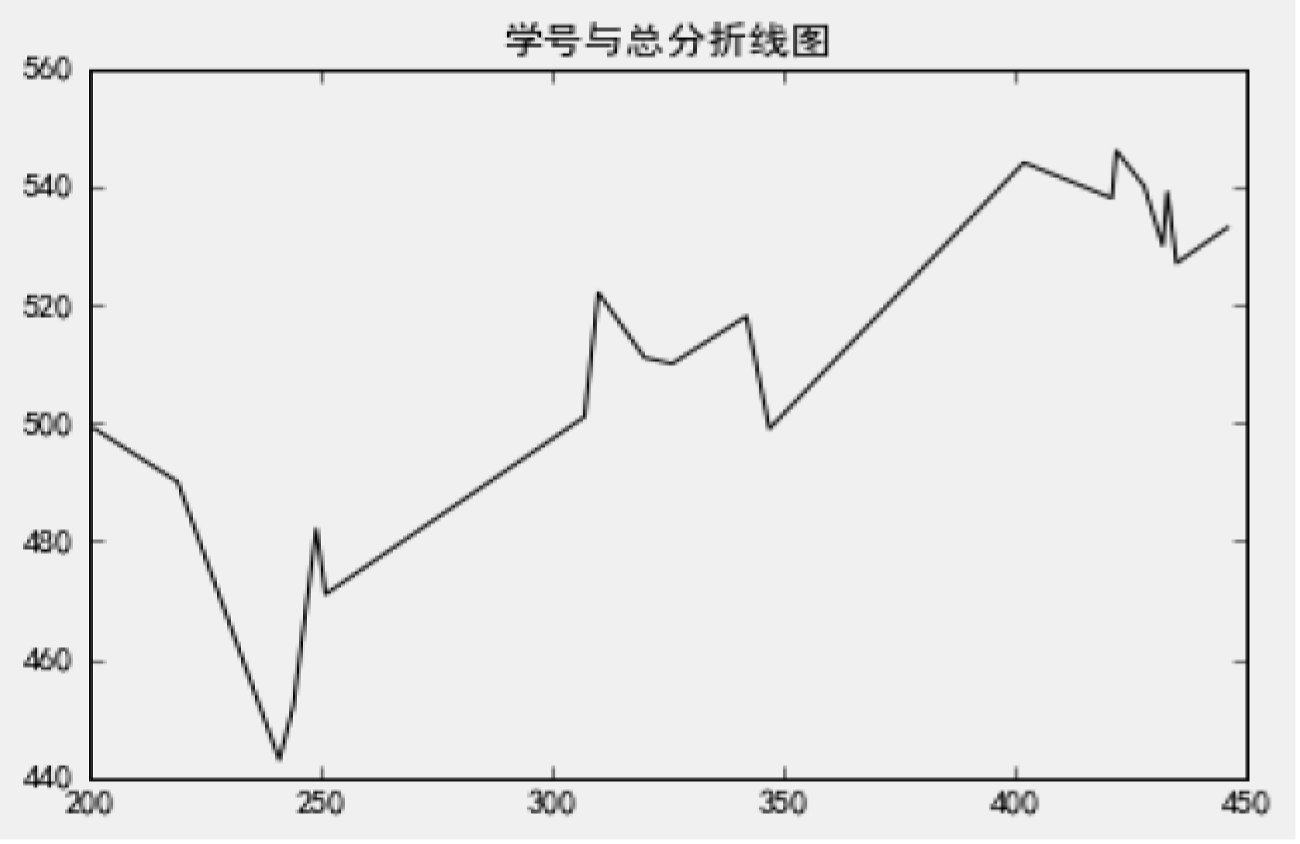


图 4-14　折线图

4.5.4　柱形图

柱形图用于显示一段时间内的数据变化或显示各项之间的比较情况，是一种单位长度的长方形，根据数据大小绘制的统计图，用来比较两个或以上的数据(时间或类别)。

涉及的主要方法如下：

```
bar(left,height,width,color)
barh(bottom,width,height,color)
```

left——x 轴的位置序列，一般采用 arange 函数产生一个序列。

height——y 轴的数值序列，也就是柱形图高度，一般就是我们需要展示的数据。

width——柱形图的宽度，一般设置为 1 即可。

color——柱形图的填充颜色。

【例 4-32】绘制柱形图：

```
import numpy
import matplotlib
from pandas import read_excel
from matplotlib import pyplot as plt
df = read_excel('e:\\rz4.xlsx',sep=',')
gb=df.groupby(by=['学号'])['总分'].agg({'总分':numpy.sum})
gb
Out[1]:
              总分
学号
2308024201  499
2308024219  490
2308024241  443
2308024244  452
2308024249  482
2308024251  471
2308024307  501
2308024310  522
2308024320  511
2308024326  510
2308024342  518
2308024347  499
2308024402  544
2308024421  538
2308024422  546
2308024428  540
2308024432  530
2308024433  539
2308024435  527
2308024446  533
index=numpy.arange(gb['总分'].size)
index
Out[2]:
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15,
16,17, 18, 19])

#为了便于图中显示中文
font = {'family':'SimHei'}
matplotlib.rc('font',**font)
#竖向柱形图
```

```
plt.title('竖向柱状图：学号-总分')
plt.bar(index,gb['总分'],1,color='G')
plt.xticks(index + 1/2,gb.index,rotation=90)
  #为了防止图中的横坐标数据重叠，选择 rotation=90
plt.show()
```

结果如图 4-15 所示。

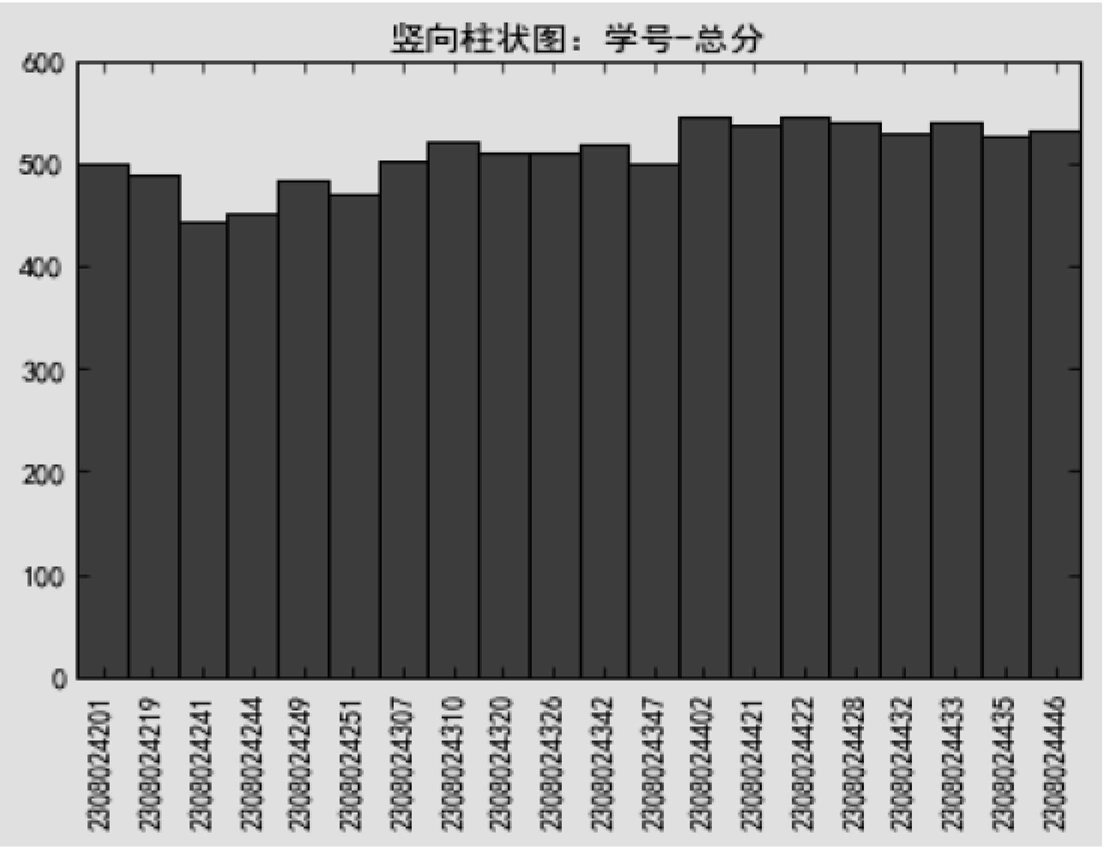


图 4-15　竖向柱形图

相应的横向柱形图代码如下：

```
#横向柱形图
plt.title('横向柱状图：学号-总分')
plt.barh(index,gb['总分'],1,color='G')
plt.yticks(index + 1/2,gb.index)
plt.show()
```

结果如图 4-16 所示。

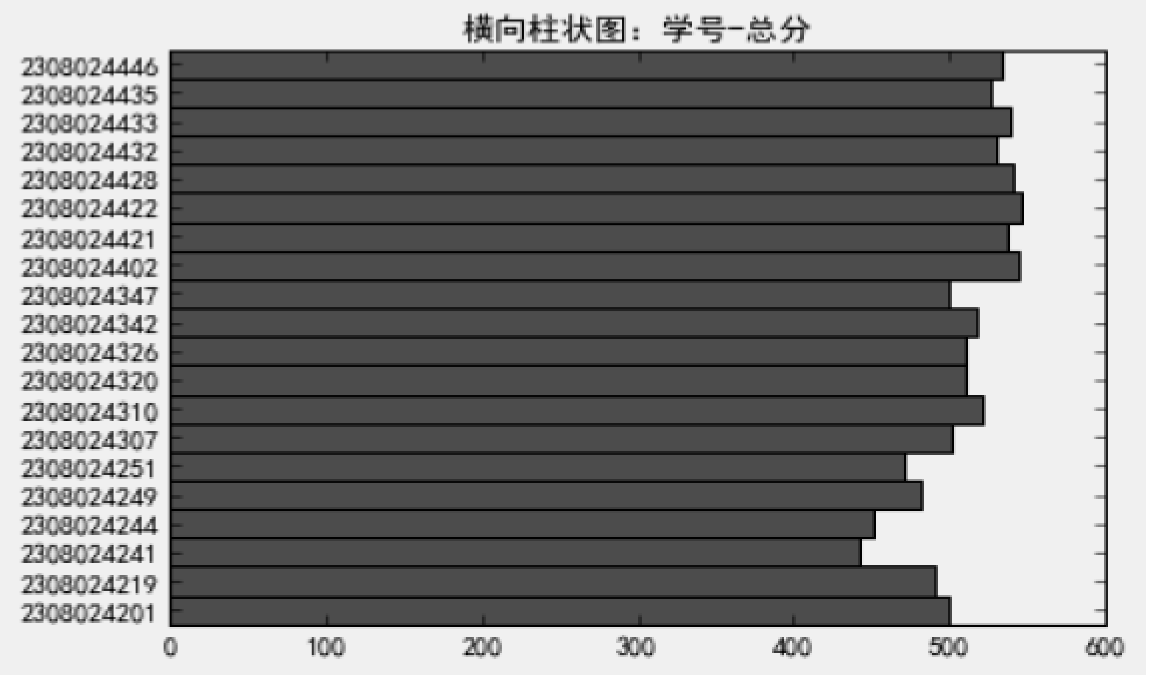


图 4-16　横向柱形图

4.5.5　直方图

直方图(Histogram)是用一系列等宽不等高的长方形来绘制的，宽度表示数据范围的间隔，高度表示在给定间隔内数据出现的频数，变化的高度形态表示数据的分布情况。

涉及的方法如下：

```
hist(x,color,bins,cumulative=False)
```

x——需要进行绘制的向量。

color——直方图填充的颜色。

bins——设置直方图的分组个数。

cumulative——设置是否累积计数，默认是 False。

【例 4-33】绘制直方图：

```
import matplotlib
from pandas import read_excel
from matplotlib import pyplot as plt

font = {'family':'SimHei'}
matplotlib.rc('font',**font)

plt.hist(df['总分'],bins=20,cumulative=True)
plt.title('总分直方图')
plt.show()
```

结果如图 4-17 所示。

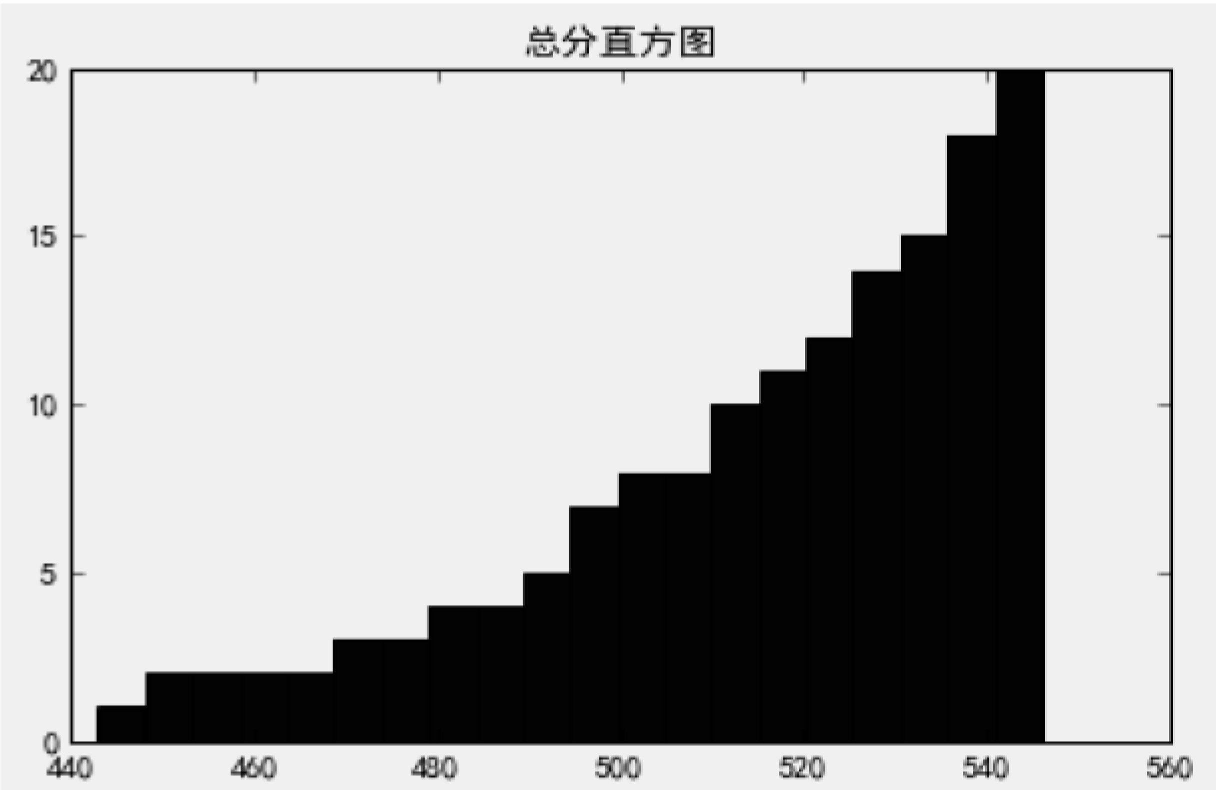


图 4-17　直方图

本 章 小 结

本章主要学习了利用 Pandas 库进行数据准备、数据处理、数据分析和数据可视化等内容。尤其是数据的整理清洗，在数据分析工作量中占到了很大的比重。如何快速地整理数据是本章的重点。

练　　习

班主任现有一班级的两张表，如下。

表一：成绩表

学号	C#	线代	Python
16010203	78	88	96
16010210	87	58	83
16010205	84	65	82
16010213	86	72	67
16010215	67	76	85
16010208	76	43	69
16010209	56	68	92
16010204	89	缺考	86
16010211	81	81	75
16010212	73	77	69
16010206	65	80	84
16010214	90	73	91
16010207	91	64	86

表二：信息表

姓名	学号	手机号码
张三	16010203	16699995521
李四	16010204	16699995522
王五	16010205	16699995523
赵六	16010206	16699995524
郑七	16010207	16699995525
钱八	16010208	16699995526
张千	16010209	16699995527
赵六	16010210	16699995528

续表

姓名	学号	手机号码
李矛	16010211	16699995529
张白	16010212	16699995510
白九	16010213	16699995511
冀二	16010214	16699995512
余一	16010215	16699995513

请帮助班主任做如下工作。

(1) 给成绩表加上姓名列。

(2) 给成绩表加上“总分”列，并求出总分。

(3) 增加列字段“等级”，标注每人的“优、良、中、及格、差”(≥90 优，≥80 良，≥70 中，≥60 及格，＜60 差)。

(4) 计算各门课程的平均成绩以及标准差。

(5) 做一总分成绩分布图，纵坐标表示成绩，横坐标表示学号，画出总分的均分横线，让每位同学的总分圆点分布在均分线上下，以观察每位同学的成绩离开均分的距离。

第 5 章

其　他

5.1 文件读写操作

Python 提供了必要的函数和方法进行默认情况下的文件基本操作。

如读写文件：

```
f = open('D:\\aa.txt')          #打开 aa.txt 文件
content = f.read()              #读取 aa.txt 文件的内容
f = open('D:\\aa.txt','a+')    #'a+'指打开 aa.txt 并在文件尾可续写
f.write('www.i-nuc.com;\n\t 爱中北'+'\n')
  #写入内容，写完换行或者加一行语句。f.write('\n')也能实现写完之后换行
f.close()       #及时关闭文件
```

打开一个文件的不同模式如表 5-1 所示。

表 5-1 打开文件的各种模式

模 式	描 述
r	打开一个文件为只读模式，文件指针位于该文件的开头。这是默认模式
rb	打开一个文件，只能以二进制格式读取，文件指针置于该文件的开头
r+	打开用于读取和写入的文件，文件指针将会在文件的开头
rb+	打开用于读取和写入二进制格式的文件，文件指针将会在文件的开头
w	打开一个文件，只写，如果该文件存在，则覆盖该文件；如果该文件不存在，则创建一个新文件用于写入
wb	打开一个文件，只能以二进制格式写入，如果该文件存在，则覆盖该文件；如果该文件不存在，则创建一个新文件用于写入
w+	打开用于写入和读取的文件，如果文件存在，则覆盖现有的文件；如果该文件不存在，则创建一个新文件用于写入
wb+	打开用于写入和读取的二进制格式的文件，如果文件存在，则覆盖现有的文件；如果该文件不存在，则创建一个新文件，用于写入
a	打开追加文件，文件指针是在文件的结尾，也就是说，该文件处于附加模式。如果该文件不存在，则创建一个新文件，用于写入
ab	打开追加的二进制格式的文件，文件指针在该文件的结尾，也就是说，该文件为追加模式；如果该文件不存在，则创建并写入一个新的文件
a+	打开为追加和读取的文件，文件指针在该文件的结尾，该文件将为追加模式；如果该文件不存在，则创建一个新文件，并读取和写入该新文件
ab+	打开一个追加和读取的二进制格式的文件，文件指针在该文件的结尾，该文件将为追加模式；如果该文件不存在，则创建一个新文件，并读取和写入该新文件
b	以二进制的形式打开文件

5.1.1　文件的读写方法

文件对象提供了三个“读”方法：.read()、.readline()和.readlines()。每种方法可以接受一个变量，以限制每次读取的数据量，但它们通常不使用变量。.read()每次读取整个文件，它通常用于将文件内容放到一个字符串变量中。然而.read()生成文件内容最直接的字符串表示，如果文件大于可用内存，则不可能实现这种处理。.readline()和.readlines()之间的差异是后者一次读取整个文件。像.read()一样，.readlines()自动将文件内容分析成一个行的列表，该列表可以由 Python 的 for ... in ... 结构进行处理。另一方面，.readline()每次只读取一行，通常比.readlines()慢得多，仅当没有足够内存可以一次读取整个文件时，才使用.readline()。

- F.read([size])：size 为读取的长度，以 byte 为单位，将文件读入 F 作为一个整体字符串。
- F.readline([size])：读一行，每操作一次读取一行，如果定义了 size，有可能返回的只是行的一部分；同 F.next()方法。
- F.readlines([size])：把文件每一行作为一个 list 的一个成员，并返回这个 list。其实它的内部是通过循环调用 readline()来实现的。如果提供 size 参数，size 是表示读取内容的总长，也就是说，可能只读到文件的一部分。
- F.write(str)：把 str 写到文件中，但 write()并不会在 str 后加上一个换行符。
- F.writelines(seq)：把 seq 的内容全部写到文件中。这个函数也只是机械地写入，不会在每行后面加上任何东西。

当读取很大的文件时，常用 fileinput 模块：

```
import fileinput
for line in fileinput.input('D:\\aa.txt'):
    print(line)
```

还可以直接使用 for，也是常用的模式之一：

```
f = open('D:\\aa.txt')
for line in f:
    print(line)
```

方法很多，还可以列表解析——使用行函数(列表函数)：

```
[line for line in open('D:\\aa.txt')]
```

使用 open 打开文件后，一定要记得调用 close()方法关闭文件。比如可以用 try-finally 语句来确保最后能关闭文件：

```
file_object = open('thefile.txt')
try:
    all_the_text = file_object.read()
finally:
    file_object.close()
```

注意： 不能把 open 语句放在 try 块里，因为当打开文件出现异常时，文件对象 file_object 无法执行 close()方法。

5.1.2 文件的其他方法

文件的其他方法说明如下。

- F.close()：关闭文件。Python 会在一个文件不用后自动关闭文件，不过这一功能没有保证，最好还是养成“手动”关闭的习惯。如果一个文件在关闭后还对其进行操作，会产生 ValueError。
- F.flush()：把缓冲区的内容写入硬盘。
- F.fileno()：返回一个长整型的“文件标签”。
- F.isatty()：文件是否是一个终端设备文件(Unix 系统中的)。
- F.tell()：返回文件操作标记的当前位置，以文件的开头为原点。
- F.next()：返回下一行，并将文件操作标记位移到下一行。当我们把一个 file 用于 for ... in file 这样的语句时，就是调用 next()函数来实现遍历的。
- F.seek(offset[,whence])：将文件操作标记(指针)移到 offset 的位置，offset 一般是相对于文件开头来计算的，一般为正数。但如果提供了 whence 参数，就不一定了，whence 可以为 0，表示从头开始计算，为 1 表示以当前位置为原点计算，为 2 表示以文件末尾为原点计算。需要注意，如果文件以 a 或 a+的模式打开，每次进行写操作时，文件操作标记会自动返回到文件末尾。可以使用 F.tell()查询指针当前的位置。
- F.truncate([size])：把文件裁成规定的大小，默认的是裁到当前文件操作标记的位置。如果 size 比文件的大小还要大，依据系统的不同，可能是不改变文件，也可能是用 0 把文件补到相应的大小，也可能是以一些随机的内容加上去。

5.1.3 文件的存储和读取

1. pickle

先看个例子：

```
import pickle
a=[1,2,3,4,5]
f=open('test01.dat','wb')   #将以 wb 格式打开(没有则新建)文件
pickle.dump(a,f)             #将文件 a 存入 f 中
f.close()
```

这个过程叫文件序列化，面对较大对象时，建议 dump()使用参数 True，能够节省不少空间，即上例中可以改为 pickle.dump(a,f,True)。

文件存进去了，还需要能读取出来，继续看例子：

```
f=open('test01.dat','rb')    #将以 rb 格式打开或新建一个文件
d=pickle.load(f)             #从 f 中读取
f.close()

d
[1, 2, 3, 4, 5]
```

2．shelve

pickle 可以完成一些简单的存取工作，但对更复杂的工作，还是有点“力不从心”。于是就有了 shelve，shelve 的操作有点像字典，更接近于数据库。例如：

```
import shelve
s=shelve.open('test02.db') #打开或新建一个文件
s['name']='yubg'             #给文件键赋值
s['sex']='man'
s['age']=40
s['demo']=['He is a teacher.']
s.close()

s=shelve.open('test02.db')
print(s['name'])      #读取 s 的全部内容时可以使用 for 来遍历

for i in s:
   print(i+":",s[i])

demo: ['He is a teacher. ']
name: yubg
sex: man
age: 40
```

但需要注意，如果要给 demo 添加一些内容，可以这样做：

```
import shelve
s = shelve.open('test02.db')
```

```
s['demo'].append('and he teaches math.')
print(s['demo'])

['He is a teacher.']
```

从内容上来看，好像没有添加进来啊？猜对了！要想添加成功，务必在打开文件的时候多添加个参数：writeback=True。例如：

```
import shelve
s = shelve.open('test02.db',writeback=True)
s['demo'].append('and he teaches math.')
print(s['demo'])
s.close()

['He is a teacher.', 'and he teaches math.']
```

说明： 在增加和删除以及查询时，都要看其类型，如 demo 是 string，则 append 是添加不成功的，因为 str 类型就不允许添加。本例中是 list，所以添加成功。当然，如果删除 name 这个 key，可以用 del，但是 demo 就不可以了，只能用.pop()。

```
s = shelve.open('test02.db')
del s['name']

for i in s:
    print(i+":",s[i])

sex: man
demo: ['He is a teacher.']
age: 40
```

5.2 with 语句

with 语句适用于对资源进行访问的场合，确保不管使用过程中是否发生异常，都会执行必要的“清理”操作，释放资源。比如文件使用后自动关闭、线程锁的自动获取和释放等。例如：

```
f = open('D:\\aa.txt')
try:
    content = f.read()
finally:
    f.close()
```

这段代码太冗长了，with 有更优雅的语法，可以很好地处理上下文环境产生的异常。下面是 with 版本的代码，自动帮我们关闭文件：

```
with open("/tmp/foo.txt") as f:
     data = f.read()
```

比较下面两段程序代码。

代码一：

```
with open(r'fileName') as f:
   for line in f:
      print(line)
```

代码二：

```
f = open(r'fileName')
try:
   for line in f:
      print(line)
finally:
   f.close()
```

比较起来，代码一优于代码二，使用 with 语句还可以减少编码量。再如：

```
with open(r'd:/aa.txt') as f:
   for line in f:
      print(line)
```

以上三行代码主要实现了以下四项工作。① 打开 D 盘文件 aa.txt；② 将文件对象赋值给 f；③ 将文件所有行输出；④ 无论代码中是否出现异常，Python 都会关闭这个文件，不必关心这些细节。

5.3　Anaconda 下安装 statsmodels 包

statismodels 是一个 Python 包，提供一些互补 scipy 统计计算功能，包括描述性统计和统计模型估计和推断。但是，Anaconda 却并不包含 statismodels 包，需要我们自己来安装。

若已经安装了 Anaconda，则从开始菜单中选择 Windows 系统中的“命令提示符”命令，如图 5-1 所示。



图 5-1　菜单中命令提示符

在弹出的命令提示符窗口中，输入如下命令，并且按 Enter 键：

```
Conda install statsmodels
```

等待安装，如图 5-2 所示。

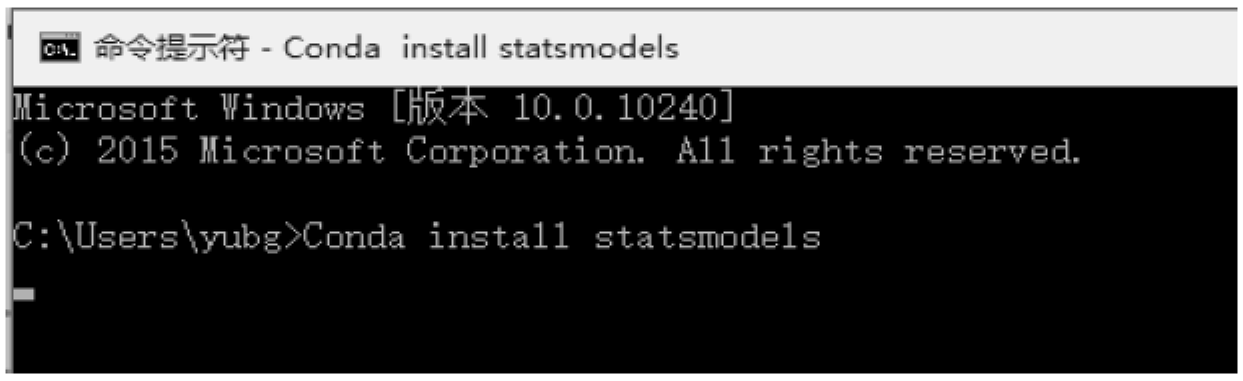


图 5-2　安装 Python 包文件

如果发现 Anaconda 有升级的文件，根据提示输入 y 即可升级，如图 5-3 和图 5-4 所示。

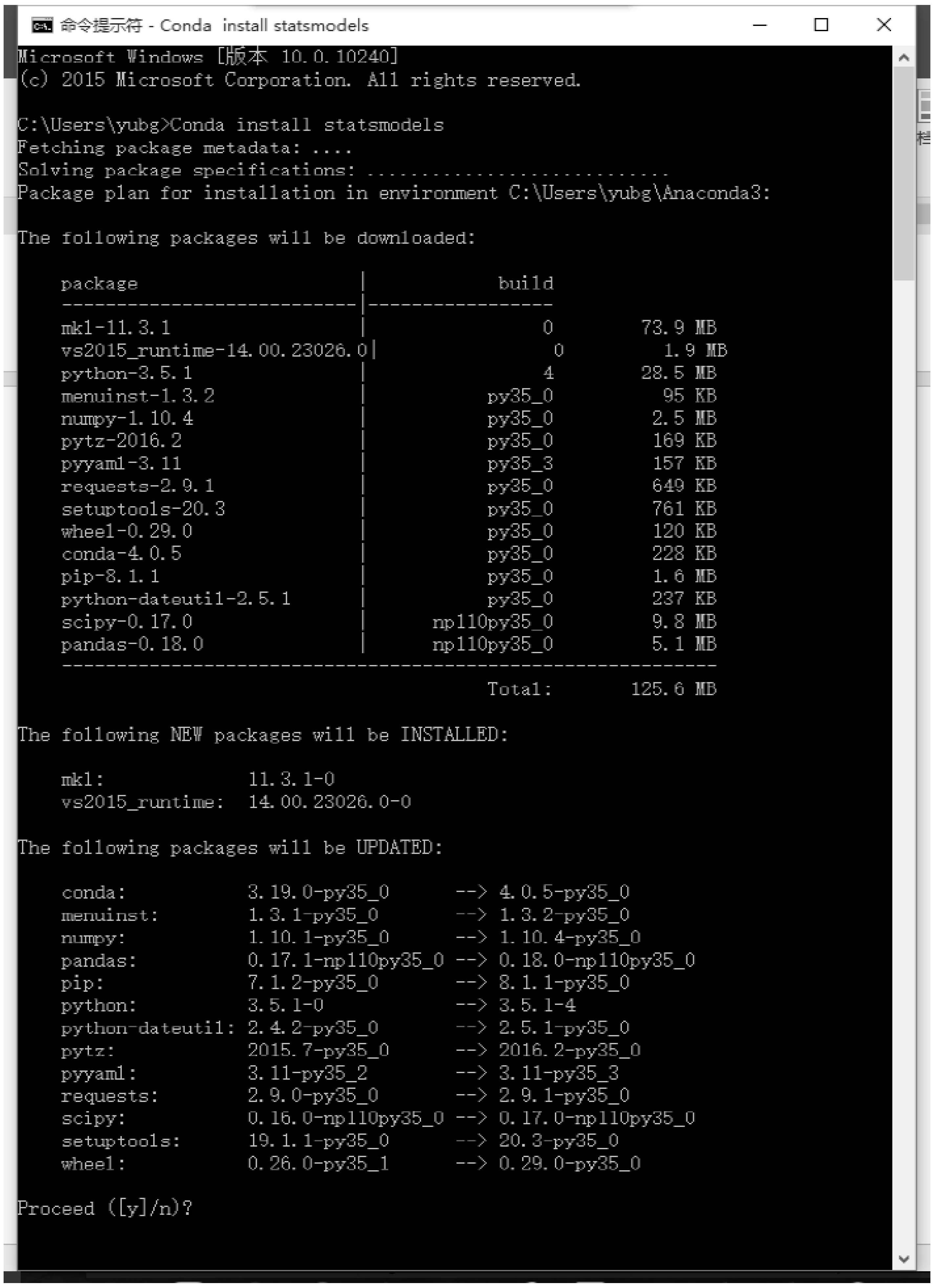


图 5-3　包文件安装截图

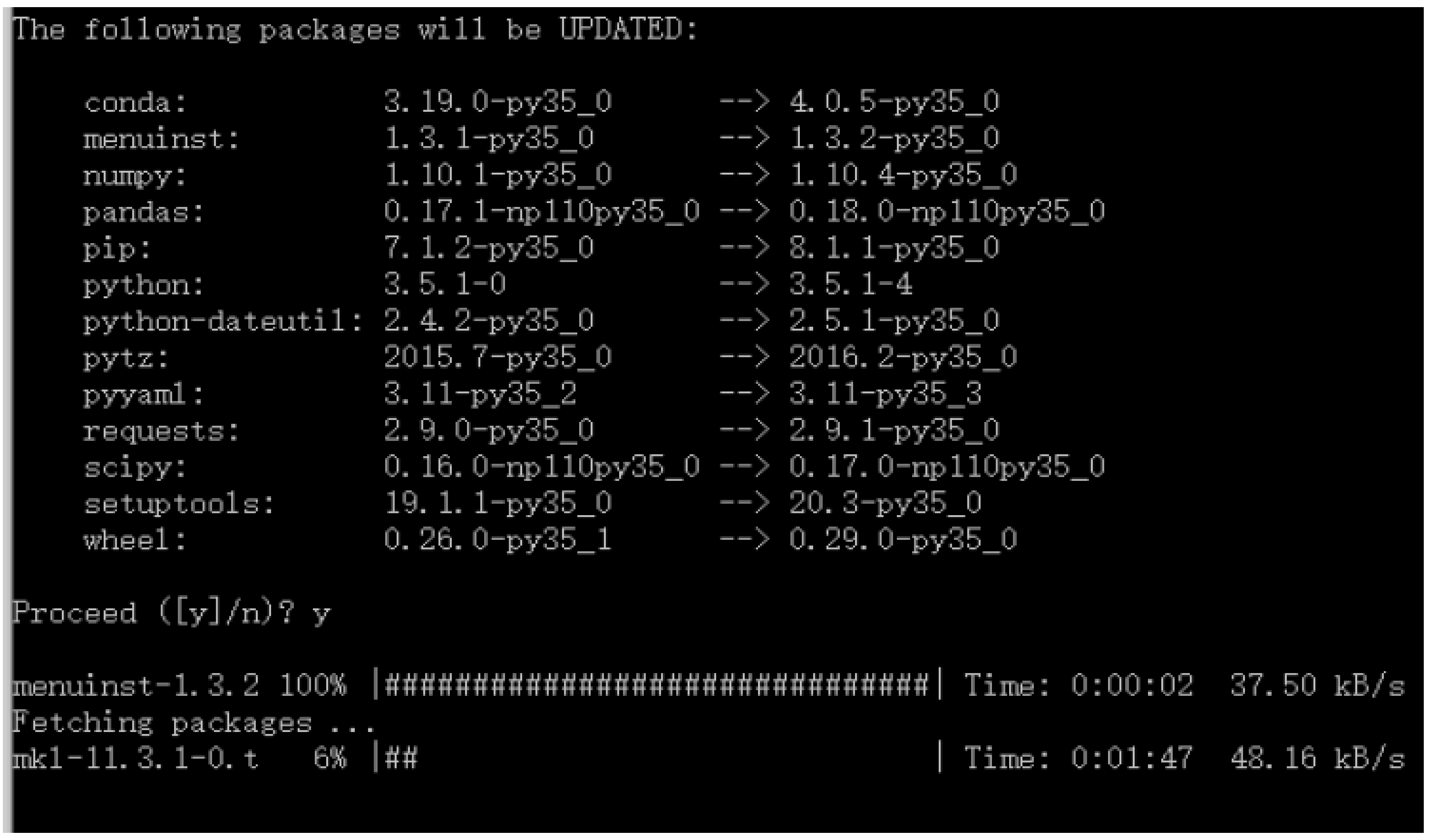


图 5-4　包文件升级截图

5.4　关于 Spyder 界面恢复默认状态的处理

在使用 Spyder 的时候，由于鼠标操作不当，很可能会将 Spyder 默认界面(图 5-5)的工作区域打乱，甚至找不到功能区。

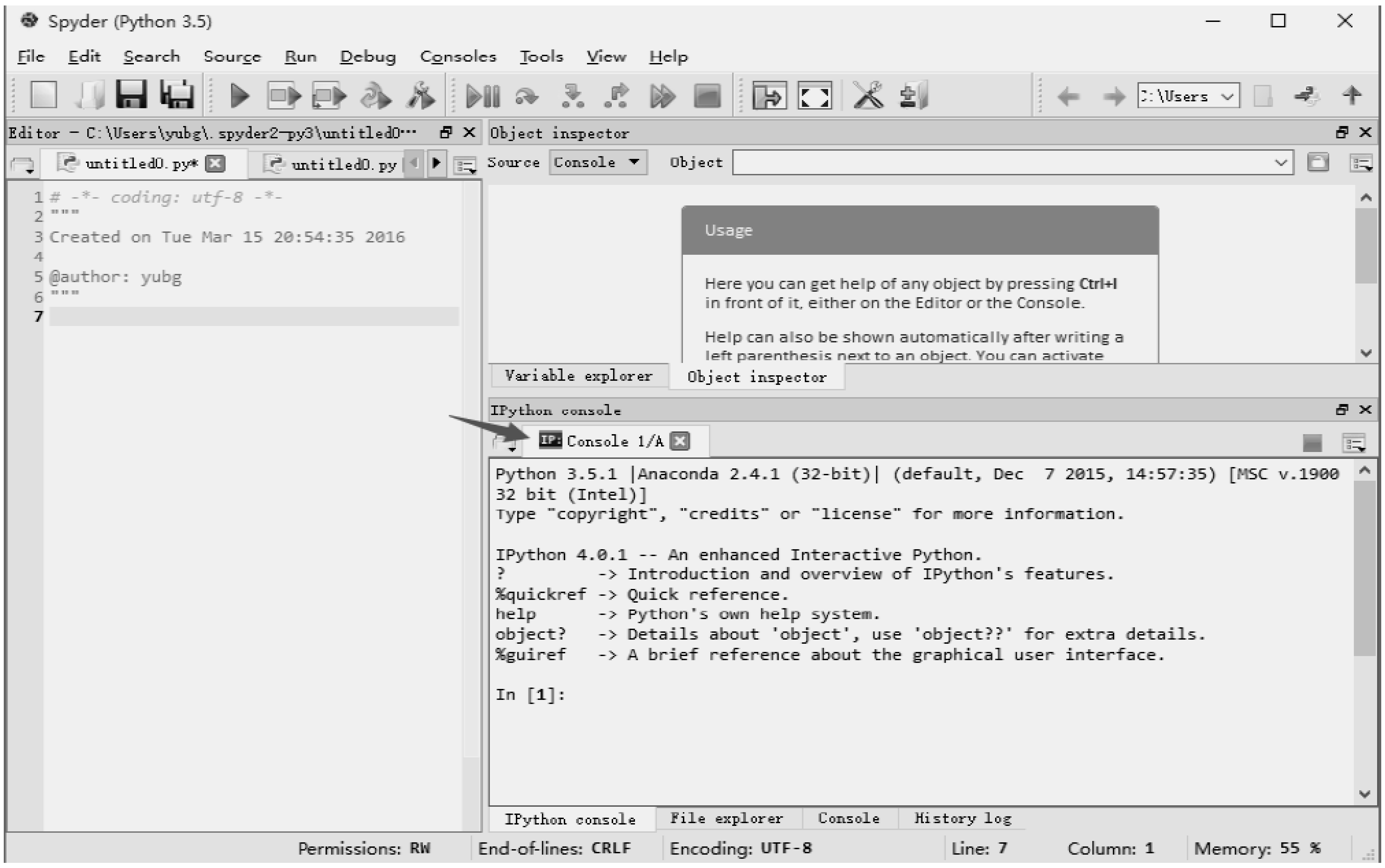


图 5-5　Spyder 系统默认的界面

常见的处理方法是选择菜单栏中的 View→Panes 命令，然后在出现的界面中勾选需要的相关功能显示项。

图 5-6 所示是一个被拖动打乱的界面，我们拟恢复原来系统默认的界面。

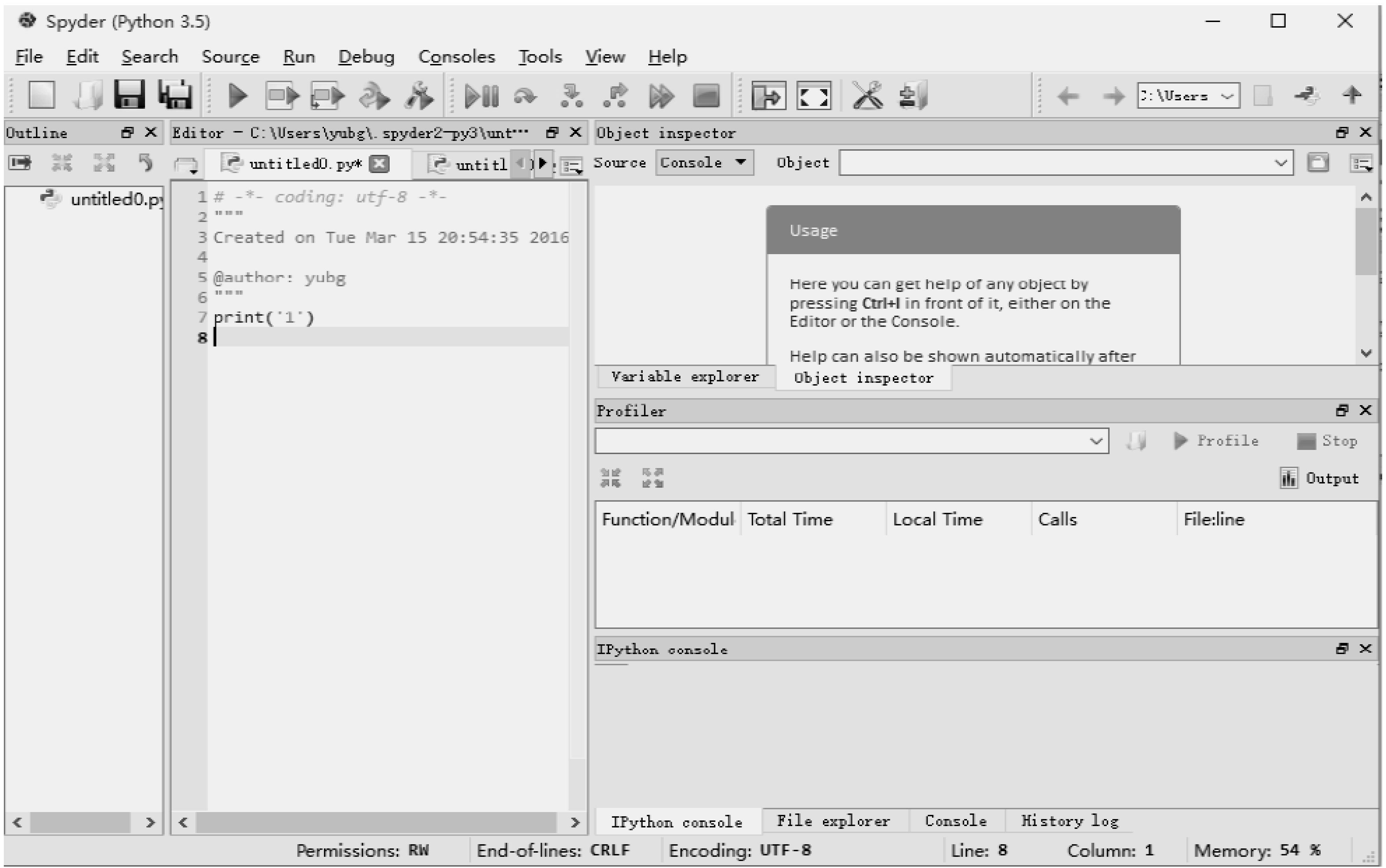


图 5-6　打乱了的 Spyder 界面

从菜单栏中选择 View→Panes 命令，如图 5-7 所示。

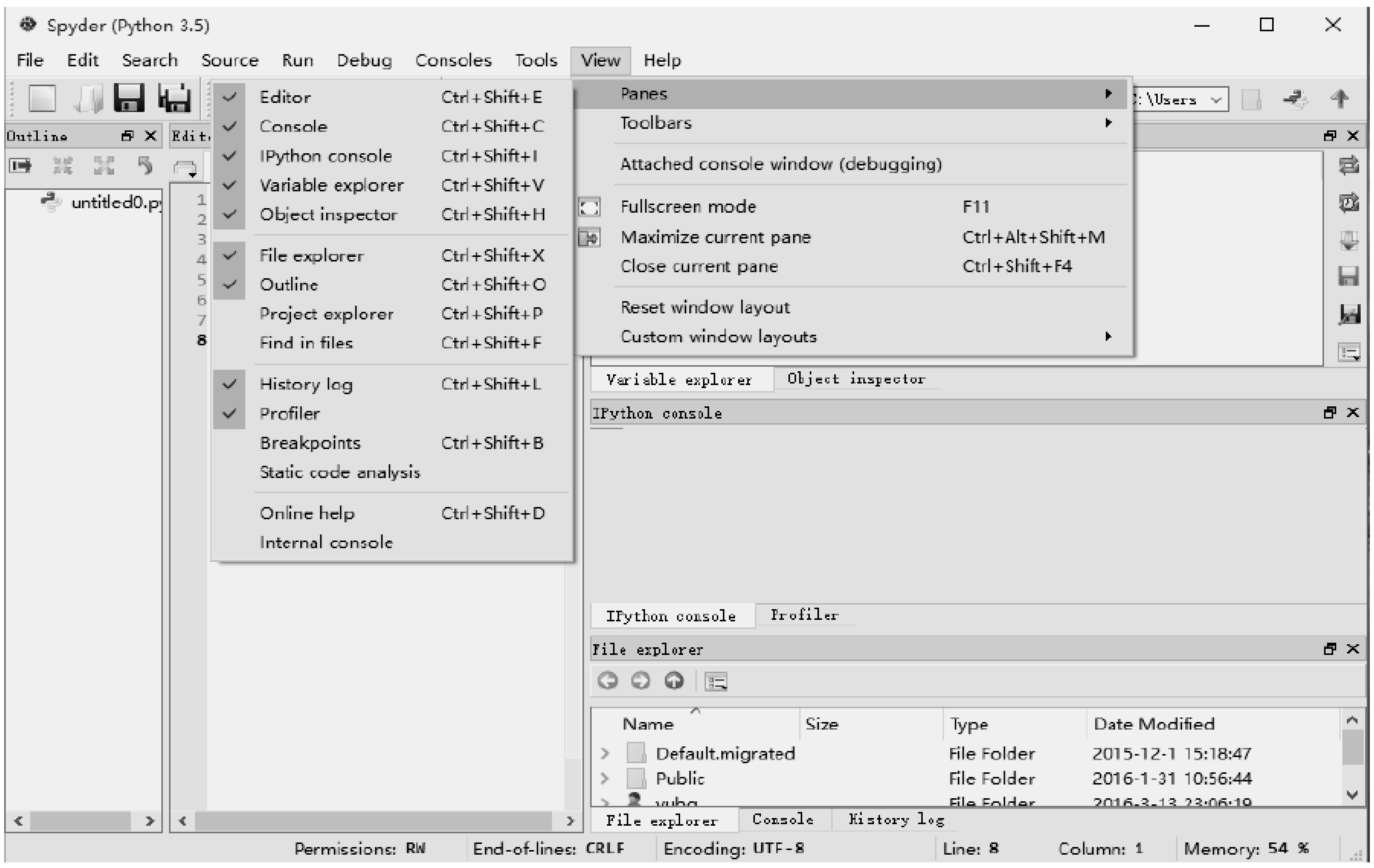


图 5-7　Spyder 中的菜单命令

另一个方法是打开 Windows 的“开始”菜单，找到 Anaconda3 程序，单击打开折叠菜

单，选择 Reset Spyder Settings 命令，如图 5-8 所示。大概等待三五秒钟，程序将自动运行并结束。当再次打开 Spyder 时，界面已经恢复到初装时的默认状态。

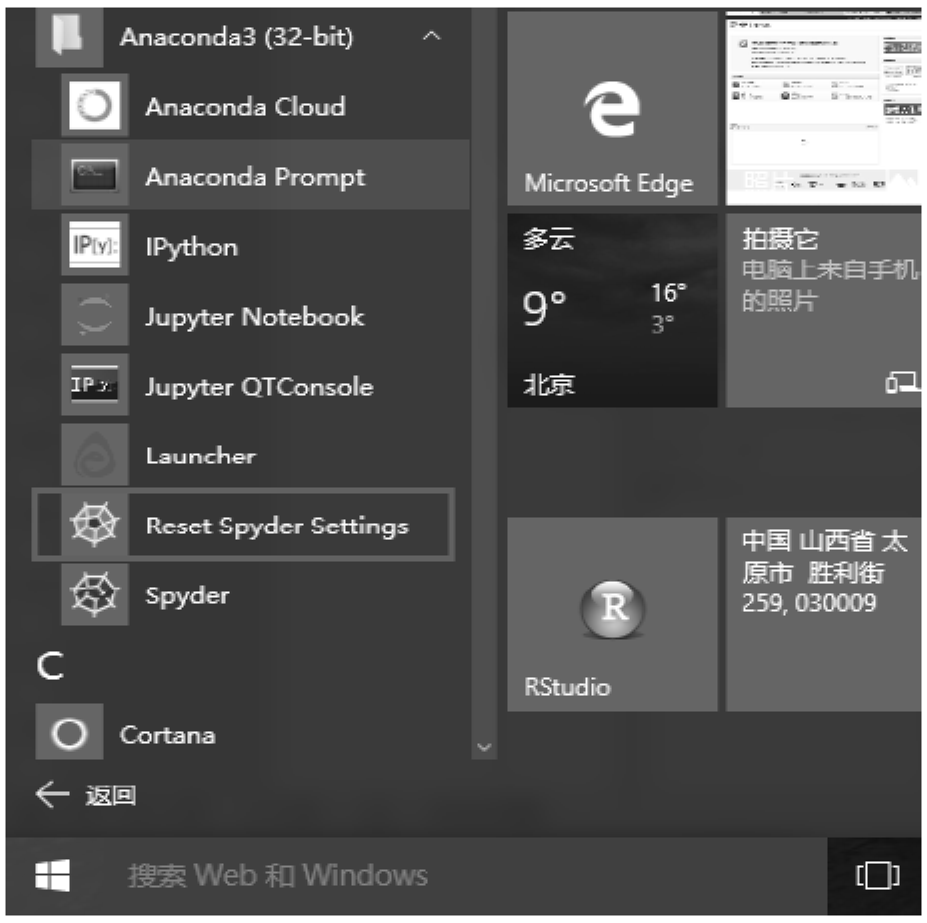


图 5-8　让 Spyder 恢复默认设置的菜单命令

5.5　关于 Python 计算精度的问题

首先看如下问题：

```
>>> a=0.1+0.1+0.1-0.3
>>> print(a)
5.551115123125783e-17
>>>
```

结果怎么会是 5.55e-17 呢？为什么不是 0？

分析：浮点数的一个普遍问题，是它们不能精确地表示十进制数，即使是最简单的数学运算也会产生小的误差。

解决方案有如下几种。

(1)　使用格式化(不推荐)：

```
>>> a = "%.30f" % (1/3)
>>> a
'0.333333333333333314829616256247'
>>>
```

可以显示，但是不准确，后面的数字往往没有意义。Python 默认的是 17 位小数的精度，当我们的计算需要使用更高的精度(超过 17 位小数)时，就需要采取特殊的方法。

(2) 使用高精度 decimal 模块，配合 getcontext：

```
>>> from decimal import *
>>> print(getcontext())
Context(prec=28, rounding=ROUND_HALF_EVEN, Emin=-999999, Emax=999999,
capitals=1, clamp=0, flags=[], traps=[InvalidOperation, DivisionByZero,
Overflow])
>>> getcontext().prec = 50
>>> b = Decimal(1)/Decimal(3)
>>> b
Decimal('0.33333333333333333333333333333333333333333333333333')
>>> c = Decimal(1)/Decimal(17)
>>> c
Decimal('0.058823529411764705882352941176470588235294117647059')
>>> float(c)
0.058823529411764705
>>>
```

decimal 的构建：可以通过整数、字符串或者元组构建 decimal.Decimal，对于浮点数，需要先将其转换为字符串。

decimal 的 context：decimal 在一个独立的 context 下工作，可以通过 getcontext 来获取当前环境。例如，前面曾经提到过，可以通过 decimal.getcontext().prec 来设定小数点精度(默认为 28)：

```
>>> from decimal import Decimal as D
>>> from decimal import getcontext
>>> getcontext()
Context(prec=6, rounding=ROUND_HALF_EVEN, Emin=-999999999,
Emax=999999999, capitals=1, flags=[Rounded, Inexact],
traps=[DivisionByZero, InvalidOperation, Overflow])

>>> getcontext().prec = 6
>>> D(1)/D(3)
Decimal('0.333333')
>>>
```

默认的 context 的精度是 28 位，可以设置为 50 位甚至更高。这样，在分析复杂的浮点数的时候，可以有更高的可控制精度。例如：

```
>>> from decimal import Decimal
>>> a = Decimal('4.2')
>>> b = Decimal('2.1')
>>> a + b
```

```
Decimal('6.3')
>>> print(a + b)
6.3
>>> (a + b) == Decimal('6.3')
True
>>>
```

注意： decimal 中的数要把它当成 str 来处理，即参加运算的数要添加单引号‘’。

(3) 使用 decimal 模块，配合 locacontext：

```
>>> from decimal import localcontext
>>> a = Decimal('1.3')
>>> b = Decimal('1.7')
>>> print(a / b)
0.7647058823529411764705882353
>>> with localcontext() as ctx:
    ctx.prec = 3
    print(a / b)

0.765
>>> with localcontext() as ctx:
    ctx.prec = 50
    print(a / b)

0.76470588235294117647058823529411764705882352941176
>>>
```

总地来说，decimal 模块主要用在涉及到金融的领域。在这类程序中，哪怕是一点小小的误差，在计算过程中都是不允许的。因此，decimal 模块为解决这类问题提供了方法。

当 Python 和数据库打交道时，也通常会遇到 decimal 对象，并且通常也是在处理金融数据时。

在有些浮点数计算问题上，可以利用 math 模块来解决，例如：

```
>>> nums = [1.23e+18, 1, -1.23e+18]
>>> sum(nums)          # 注意结果为什么不是 1?
0.0
>>>
```

上面的错误可以利用 math.fsum()所提供的更精确计算能力来解决：

```
>>> import math
>>> math.fsum(nums)
1.0
```

```
>>>
```

Math 模块提供了以下功能函数：

```
>>> import math
>>> dir(math)
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos',
'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign',
'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs',
'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot',
'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log',
'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'sin',
'sinh', 'sqrt', 'tan', 'tanh', 'trunc']
>>>
```

5.6 矩 阵 运 算

5.6.1 创建矩阵

矩阵的创建可以利用 numpy 包，如创建矩阵 A 和 B：

```
>>> import numpy as np
>>> A = np.mat([[1,2,3],[4,5,6]])
>>> A
matrix([[1, 2, 3],
       [4, 5, 6]])

>>> B= [[1,2],[3,4]]
>>> B= np.array(B)
>>> B
array([[1, 2],
      [3, 4]])
>>>
```

在 numpy 里，mat 是 matrix 的一个别名。

5.6.2 矩阵属性

a.T：返回自身的转置(np.array(B)构成的矩阵仅可用此属性)。

a.H：返回自身的共轭转置。

a.I：返回自身的逆矩阵。

a.A：返回自身数据的二维数组的一个视图(没有做任何的拷贝)。

例如：

```
import numpy as np
a.transpose()                  #返回行列转置，等同于 a.T
a.trace()                      #计算矩阵 a 的迹
np.trace(a)                    #计算矩阵 a 的迹
np.linalg.inv(a)               #矩阵 a 的逆矩阵
np.linalg.matrix_rank(a)       #求矩阵的秩
np.dot(A,B)                    #矩阵 A、B 的乘积
np.linalg.det(a)               #返回的是矩阵 a 的行列式
c,d = np.linalg.eig(a)         #矩阵 a 的特征值 c 和特征向量 d
np.linalg.norm(a,ord=None)     #计算矩阵 a 的范数
np.linalg.cond(a,p=None)       #矩阵 a 的条件数
a.shape                        #可以获取矩阵的大小

>>> from numpy import *
>>> a=mat('1 2 3; 4 5 3')
>>> print((a*a.T).I)

[[0.29239766  -0.13450292]
[-0.13450292   0.08187135]]

#求矩阵的秩：
>>> import numpy
>>> i=numpy.eye(4)
>>> i
array([[ 1.,  0.,  0.,  0.],
       [ 0.,  1.,  0.,  0.],
       [ 0.,  0.,  1.,  0.],
       [ 0.,  0.,  0.,  1.]])
>>> numpy.linalg.matrix_rank(i)
4
>>>
```

5.6.3　解线性方程组

求线性方程组 AX=B 的解：

```
np.linalg.solve(A,B)
```

例如：

```
A=np.mat([[1,2],[3,4]])
B=np.mat([[5,6]]).T
```

```
s=np.linalg.solve(A,B)

matrix([[-4. ],
        [ 4.5]])
```

5.6.4 线性规划最优解

看一个例子：小明找到了一份实习工作，于是想租一个房子，最好离公司近点，但是还没毕业，学校时不时还有事，所以不能离学校太远；而且有时还要去女朋友那里，她希望小明就住在她附近，于是小明该如何选择房子的地址？

具体我们假定公司、学校、女友在地图上的坐标分别是(1,1)、(4,6)、(9,2)，求小明的租房坐标。

这里需要使用 scipy 提供的 scipy.optimize.mininize 方法，首先需要设计一个计算距离的方程：

```
import numpy as np
from scipy.optimize import minimize
#租房到公司、学校、女友的距离平方和最小方程，房子的坐标:[coord[0],coord[1]]
def f(coord,x,y):
   return np.sum((coord[0]-x)**2+(coord[1]-y)**2)

#把公司、学校、女友三地的坐标保存在两个向量里
x= np.array([1,4,9])  #公司、学校、女友的横坐标
y= np.array([1,6,2])  #公司、学校、女友的纵坐标

#找一个起始点，并看看租房到三地的距离平方和
initial = np.array([50,5])  #随便选一个租房地址作为起点
print(f(initial,x,y))

#求最优解并打印
res = minimize(f,initial,args=(x,y)) #最优化
print(res.x)      #打印最优的房子坐标
print(f(res.x,x,y))

6224
[ 4.66666667  3.00000001]
46.6666666667
```

绘制出地图，标注上租房的点 house(如图 5-9 所示)：

```
import matplotlib.pyplot as plt
labels = ['company','school','girl']
```

```
plt.scatter(x,y) #按照 x、y 画图

for i in range(3):
    plt.arrow(res.x[0],res.x[1],-res.x[0]+x[i],
         -res.x[1]+y[i],head_length=-0.1,head_width=0.1,fc='k')
    plt.text(x[i],y[i],labels[i])

plt.text(res.x[0],res.x[1],'house')
plt.show()
```

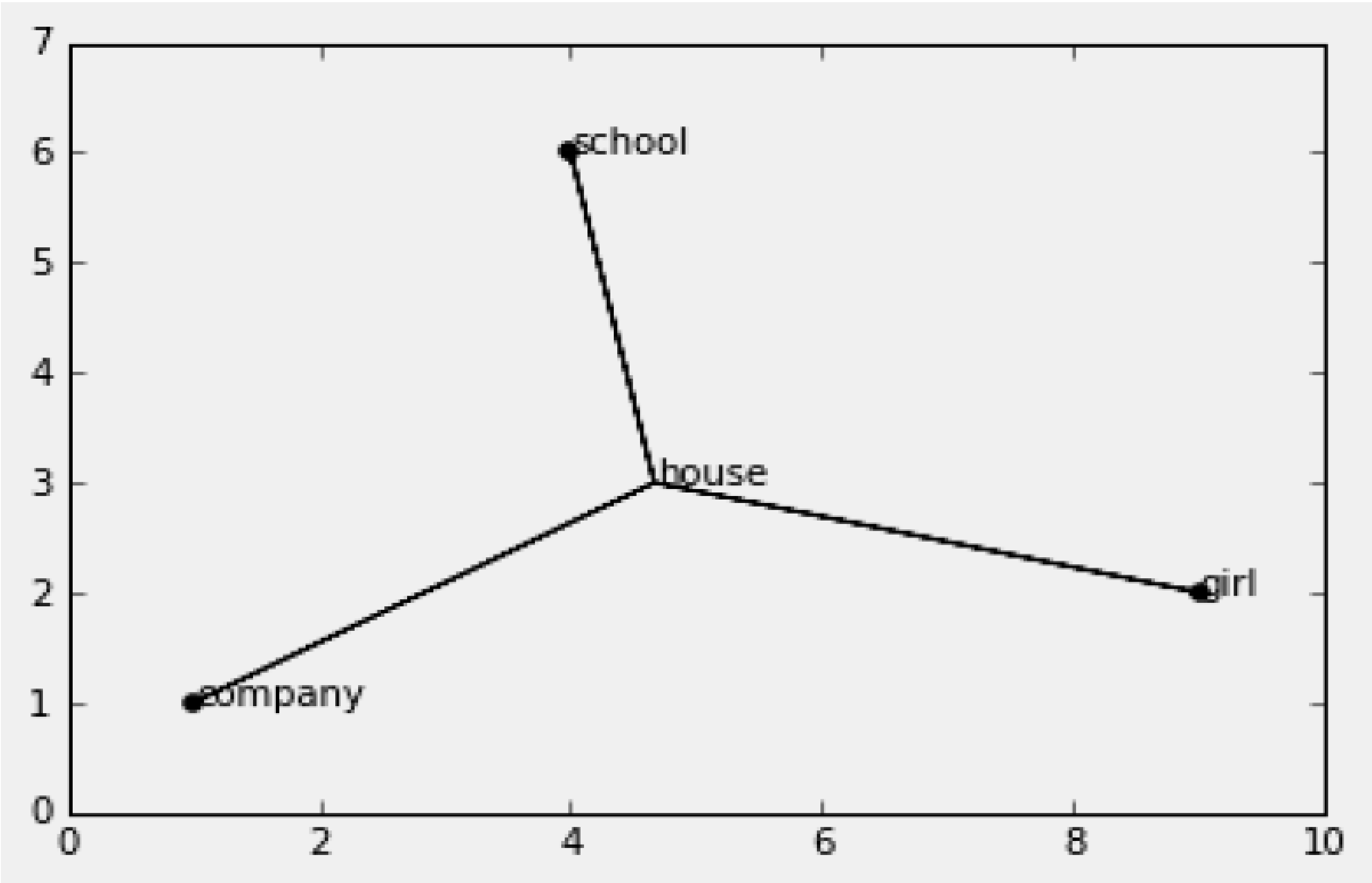


图 5-9　租房地点 house 与其他三地的位置

5.7　正则表达式

字符串是编程时涉及到的最多的一种数据结构，对字符串进行操作的需求几乎无处不在。比如判断一个字符串是否为合法的 E-mail 地址，虽然可以编程提取@前后的子串，再分别判断是否是单词和域名，但这样做不但麻烦，而且代码难以复用。

正则表达式是一种用来匹配字符串的强有力的工具。它的设计思想是用一种描述性的语言来给字符串定义一个规则，凡是符合规则的字符串，我们就认为它“匹配”了，否则，该字符串就是不合法的。

所以，判断一个字符串是否为合法的 E-mail 的方法如下。

(1)　创建一个匹配 E-mail 的正则表达式。

(2)　用该正则表达式去匹配用户的输入，判断是否合法。

因为正则表达式是用字符串表示的，所以要首先了解如何用字符来描述字符。

1．准备知识

在正则表达式中，如果直接给出字符，就是精确匹配。用\d 可以匹配一个数字，用\w 可以匹配一个字母或数字，用英文句点可以匹配任意字符，所以：

‘00\d’——可以匹配‘007’，但无法匹配‘00A’，也就是说，‘00’后面只能是数字。

‘\d\d\d’——可以匹配‘010’，只可匹配三位数字。

‘\w\w\d’——可以匹配‘py3’，前两位可以是数字或者字母，但是第三位只能是数字。

‘py.’——可以匹配‘pyc’、‘pyo’、‘py!’等。

在正则表达式中，用*表示任意个字符(包括 0 个)，用+表示至少一个字符，用?表示 0 个或 1 个字符，用{n}表示 n 个字符，用{n,m}表示 n 到 m 个字符。

下面看一个复杂的例子：\d{3}\s+\d{3,8}。

从左到右解读如下。

(1) \d{3}表示匹配 3 个数字。例如‘010’。

(2) \s 可以匹配一个空格(也包括 Tab 等空白符)，所以\s+表示至少有一个空格。例如匹配‘ ’、‘   ’等。

(3) \d{3,8}表示 3~8 个数字。例如‘1234567’。

综合起来，上面的正则表达式可以匹配以任意个空格隔开的区号为 3 个数字、号码为 3~8 个数字的电话号码。如‘021   8234567’。

如果要匹配‘010-12345’这样的号码呢？由于‘-’是特殊字符，在正则表达式中，要用‘\’转义，所以正则式应该是\d{3}\-\d{3,8}。

但是，仍然无法匹配‘010 - 12345’，因为这里‘-’两侧带有空格。所以需要更复杂的匹配方式。

2．进阶

要做更精确的匹配，可以用[]表示范围，比如：

[0-9a-zA-Z_]——可以匹配一个数字、字母或者下划线。

[0-9a-zA-Z_]+——可以匹配至少由一个数字、字母或者下划线组成的字符串，比如‘a100’、‘0_Z’、‘Py3000’等。

[a-zA-Z_][0-9a-zA-Z_]*——可以匹配由字母或下划线开头，后接任意个由一个数字、字母或者下划线组成的字符串，也就是 Python 合法的变量。

[a-zA-Z_][0-9a-zA-Z_]{0, 19}——更精确地限制了变量的长度是 1~20 个字符(前面 1 个字符+后面最多 19 个字符)。

A|B 可以匹配 A 或 B，所以(P|p)ython 可以匹配'Python'或者'python'。

^表示行的开头，^\d 表示必须以数字开头。

$表示行的结束，\d$表示必须以数字结束。

应注意，py 也可以匹配'python'，但加上^py$就变成了整行匹配，就只能匹配'py'了。

具体的正则表达式常用符号见表 5-2。

表 5-2 正则表达式的常用符号

符 号	含 义	例 子	匹配结果
*	匹配前面的字符、表达式或括号里的字符 0 次或多次	a*b*	aaaaaaa、aaaaabbb
+	匹配前面的字符、表达式或括号里的字符至少一次	a+b+	aabbb、abbbbb、aaaaab
?	匹配前面的一次或 0 次	Ab?	A、Ab
.	匹配任意单个字符，包括数字、空格和符号	b.d	bad、b3d、b#d
[]	匹配[]内的任意一个字符，即任选一个	[a-z]*	zero、hello
\	转义符，把后面的特殊意义的符号按原样输出	\.|\\	.|\
^	指字符串开始位置的字符或子表达式	^a	apple、aply、asdfg
$	经常用在表达式的末尾，表示从字符串的末端匹配，如果不用它，则每个正则表达式的实际表达形式都带有.*作为结尾。这个符号可以看成^符号的反义词	[A-Z]*[a-z]*$	ABDxerok、Gplu、yubg、YUBEG
|	匹配任意一个由 | 分割的部分	b(i|ir|a)d	bid、bird、bad
?!	不包含，这个组合经常放在字符或者正则表达式前面，表示这些字符不能出现。如果在某整个字符串中全部排除某个字符，就要加上^和$符号	^((?![A-Z]).)*$	除了大写字母以外的所有字母字符均可：nu-here、&hu238-@
()	表达式编组，()内的正则表达式会优先运行	(a*b)*	aabaaab、aaabab、abaaaabaaaabaaab
{m,n}	匹配前面的字符串或者表达式 m 到 n 次，包含 m 和 n 次	go{2,5}gle	gooogle、goooogle、gooooogle、goooooogle
[^]	匹配任意一个不在中括号内的字符	[^A-Z]*	sed、sead@、hes#23
\d	匹配一位数字	a\d	a3、a4、a9
\D	匹配一位非数字	3\D	3A、3a、3-
\w	匹配一个字母或数字	\w	3、A、a

3. re 模块

有了准备知识，就可以在 Python 中使用正则表达式了。Python 提供 re 模块，包含所有正则表达式的功能。

由于 Python 的字符串本身也用\转义，所以要特别注意：

```
s = 'ABC\\-001'          #Python 的字符串
```

对应的正则表达式字符串变成：

```
# 'ABC\-001'
```

因此强烈建议使用 Python 的 r 前缀，就不用考虑转义的问题了：

```
s = r'ABC\-001'          #Python 的字符串
```

对应的正则表达式字符串不变：

```
# 'ABC\-001'
```

先看看如何判断正则表达式是否匹配：

```
>>> import re
>>> re.match(r'^\d{3}\-\d{3,8}$', '010-12345')
<_sre.SRE_Match object; span=(0, 9), match='010-12345'>
>>> re.match(r'^\d{3}\-\d{3,8}$', '010 12345')
>>>
```

match()方法判断是否匹配，如果匹配成功，返回一个 Match 对象，否则返回 None。常见的判断方法是：

```
test = '用户输入的字符串'
if re.match(r'正则表达式', test):
    print('ok')
else:
    print('failed')
#输出：
failed
```

4. 切分字符串

用正则表达式切分字符串比用固定的字符更灵活，请看正常的切分代码：

```
>>> 'a b   c'.split(' ')
['a', 'b', '', '', 'c']
>>>
```

执行上面代码，结果显示，无法识别连续的空格。下面运行正则表达式试一下：

```
>>> re.split(r'\s+', 'a b   c')
['a', 'b', 'c']
>>>
```

无论多少个空格都可以正常分割。下面加入“\,”试试：

```
>>> re.split(r'[\s\,]+', 'a,b, c  d')
['a', 'b', 'c', 'd']
>>>
```

再加入“\,\;”试试：

```
>>> re.split(r'[\s\,\;]+', 'a,b;; c  d')
['a', 'b', 'c', 'd']
>>>
```

如果用户输入了一组标签，可以用正则表达式把不规范的输入转化成正确的数组。

5. 分组

除了简单地判断是否匹配之外，正则表达式还有提取子串的强大功能。用()表示的即是要提取的分组(Group)。

例如^(\d{3})-(\d{3,8})$分别定义了两个组，可以直接从匹配的字符串中提取出区号和本地号码：

```
>>> m = re.match(r'^(\d{3})-(\d{3,8})$', '010-12345')
>>> m
<_sre.SRE_Match object; span=(0, 9), match='010-12345'>
>>> m.group(0)
'010-12345'
>>> m.group(1)
'010'
>>> m.group(2)
'12345'
>>>
```

如果正则表达式中定义了组，就可以在 Match 对象上用 group()方法提取出子串。

注意到 group(0)是原始字符串，group(1)、group(2)、……表示第 1、2、……个子串。

提取子串非常有用，例如：

```
>>> t = '19:05:30'
>>> m = re.match(r'^(0[0-9]|1[0-9]|2[0-3]|[0-9])\:(0[0-9]|1[0-9]|2[0-9]|3[0-9]|4[0-9]|5[0-9]|[0-9])\:(0[0-9]|1[0-9]|2[0-9]|3[0-9]|4[0-9]|5[0-9]|[0-9])$', t)
>>> m.groups()
```

```
('19', '05', '30')
>>>
```

这个正则表达式可以直接识别合法的时间。但有些时候，用正则表达式也无法做到完全验证，比如识别日期：

```
'^(0[1-9]|1[0-2]|[0-9])-(0[1-9]|1[0-9]|2[0-9]|3[0-1]|[0-9])$'
```

对于‘2-30’、‘4-31’这样的非法日期，用正则还是识别不了，或者说写出来非常困难，这时就需要程序配合识别了。

6．贪婪匹配

最后需要特别指出的是，正则匹配默认是贪婪匹配，也就是匹配尽可能多的字符。

举例如下，匹配出数字后面的 0：

```
>>> re.match(r'^(\d+)(0*)$', '102300').groups()
('102300', '')
>>>
```

由于\d+采用贪婪匹配，直接把后面的 0 全部匹配了，结果 0*只能匹配空字符串了。

必须让\d+采用非贪婪匹配(也就是尽可能少匹配)，才能把后面的 0 匹配出来，加个?就可以让\d+采用非贪婪匹配：

```
>>> re.match(r'^(\d+?)(0*)$', '102300').groups()
('1023', '00')
>>>
```

7．编译

当我们在 Python 中使用正则表达式时，re 模块内部会做两件事情。

(1) 编译正则表达式，如果正则表达式的字符串本身不合法，会报错。

(2) 用编译后的正则表达式去匹配字符串。

如果一个正则表达式要重复使用几千次，出于效率的考虑，我们可以预编译该正则表达式，接下来重复使用时，就不需要编译这个步骤了，可以直接匹配：

```
>>> import re
#编译：
>>> re_telephone = re.compile(r'^(\d{3})-(\d{3,8})$')
#使用：
>>> re_telephone.match('010-12345').groups()
('010', '12345')
>>> re_telephone.match('010-8086').groups()
('010', '8086')
```

```
>>>
```

编译后生成 Regular Expression 对象，由于该对象自己包含了正则表达式，所以调用对应的方法时，不用给出正则字符串。

5.8　使用 urllib 打开网页

Python 的 webbrowser 模块支持对浏览器进行一些操作，主要有三种方法。

代码如下：

```
import webbrowser

webbrowser.open(url, new=0, autoraise=True)
webbrowser.open_new(url)
webbrowser.open_new_tab(url)
```

首先了解 webbrowser.open()方法：

```
webbrowser.open(url, new=0, autoraise=True)
```

在系统的默认浏览器中访问 url 地址，如果 new=0，url 会在同一个浏览器窗口中打开；如果 new=1，新的浏览器窗口会被打开；new=2 时新的浏览器 tab 会被打开。而 webbrowser.get()方法可以获取到系统浏览器的操作对象。使用 webbrowser.register()可以注册浏览器类型。

例如：

```
#-*- coding:UTF-8 -*-
import webbrowser
url = 'http://www.i-nuc.com/iNUC/'
webbrowser.open(url)
print(webbrowser.get())
```

这样就可以打开一个网站页面，但使用的是默认 IE 打开的，如果想用 360 浏览器打开，方法如下：

```
import webbrowser
BrowserPath = r'C:\Users\yubg\AppData\Roaming\360se6\Application\360se.exe'
webbrowser.register('360', None,
   webbrowser.BackgroundBrowser(BrowserPath))
webbrowser.get('360').open_new_tab('http://www.i-nuc.com/iNUC/')
```

而使用 urllib 模块打开网站，整个程序只需用两行代码即可：

```
import urllib
```

```
print(urllib.request.urlopen('http://www.i-nuc.com/iNUC/').read())
```

在 Python 2.x 版本中可以直接使用 import urllib 来进行操作，但是 3.x 版本的 Python 中，使用的是 import urllib.request 来进行操作。

urlopen 方法的格式：

```
urllib.request.urlopen(url[, data[, proxies]])
```

参数 url 表示远程数据的路径，一般是网址；参数 data 表示以 post 方式提交到 url 的数据(提交数据的两种方式为 post 与 get。如果不清楚，也不必太在意，一般情况下很少用到这个参数)；参数 proxies 用于设置代理(这里不详细介绍怎样使用代理，感兴趣的读者可以去翻阅 Python 手册的 urllib 模块)。

urlopen 返回一个类文件对象，它提供了下列方法。

- *read()、readline()、readlines()、fileno()、close()：这些方法的使用方式与文件对象完全一样。
- *info()：返回一个 httplib.HTTPMessage 对象，表示远程服务器返回的头信息。
- *getcode()：返回 HTTP 状态码。如果是 HTTP 请求，200 表示请求成功完成；404 表示网址未找到。
- *geturl()：返回请求的 url。

对上面的代码进行扩充，可以运行下面的例子，加深对 urllib 的印象：

```
import urllib
print(urllib.request.urlopen('http://www.i-nuc.com/iNUC/').read())
inuc = urllib.request.urlopen('http://www.i-nuc.com/iNUC/')
print('http header:\n', inuc.info())
print('http status:', inuc.getcode())
print('url:', inuc.geturl())
for line in inuc:     #就像在操作本地文件
    print(line)
inuc.close()
```

urlretrieve 方法的格式：

```
urllib.request.urlretrieve(url[, filename[, reporthook[, data]]])
```

urlretrieve 方法直接将远程数据下载到本地。参数 filename 指定了保存到本地的路径(如果未指定该参数，urllib 会生成一个临时文件来保存数据)；参数 reporthook 是一个回调函数，当连接上服务器，以及相应的数据块传输完毕的时候，会触发该回调。可以利用回调函数来显示当前的下载进度，下面的代码将会展示。参数 data 指 post 到服务器的数据。该方法返回一个包含两个元素的元组(filename, headers)，filename 表示保存到本地的路

径，header 表示服务器的响应头。

下面通过例子来演示这个方法的使用，本例将“爱中北”首页的 HTML 抓取到本地，保存在 D:\i-nuc.html 文件中，同时显示下载的进度。程序如下：

```
def cbk(a, b, c):
    '''
    回调函数
    @a: 已经下载的数据块
    @b: 数据块的大小
    @c: 远程文件的大小
    '''
    per = 100.0 * a * b / c
    if per > 100:
      per = 100
      print('%.2f%%' % per)
url = 'http://www.i-nuc.com'
local = 'd:\\i-nuc.html'
urllib.request.urlretrieve(url, local, cbk)
```

运行结果如图 5-10 所示。



```
In [1]: import urllib
   ...: def cbk(a, b, c):
   ...:     '''
   ...:     回调函数
   ...:     @a: 已经下载的数据块
   ...:     @b: 数据块的大小
   ...:     @c: 远程文件的大小
   ...:     '''
   ...:     per = 100.0 * a * b / c
   ...:     if per > 100:
   ...:        per = 100
   ...:        print('%.2f%%' % per)
   ...:
   ...: url = 'http://www.i-nuc.com'
   ...: local = 'd:\\i-nuc.html'
   ...: urllib.request.urlretrieve(url, local, cbk)
100.00%
Out[1]: ('d:\\i-nuc.html', <http.client.HTTPMessage at 0xcf29910>)

In [2]:
```

图 5-10　运行结果

urllib 中还提供了一些辅助方法，用于对 url 进行编码、解码。

url 中不能出现一些特殊的符号，有些符号有特殊的用途。我们知道，以 get 方式提交数据的时候，会在 url 中添加 key = value 这样的字符串，所以在 value 中不允许有“=”，因此要对其进行编码；与此同时，服务器接收到这些参数的时候，要进行解码，还原成原始的数据。

5.9 网页数据抓取

这里给出用 Python 抓取一个网站跟帖回复的例子。

知乎网上有一个标题为“已经有女朋友了，但又遇到更喜欢的对象怎么办？”的帖子，里面的一些回复实在很搞笑，但是一页一页地看又有点麻烦，于是想全部“爬”下来存放到一个文件里，这样，就可以随时查看到全部的回复内容了。

本节主要使用 Python 实现简单的网络爬取，以获取帖子的全部回复。

工具：Anaconda 3。

网址：https://www.zhihu.com/question/19649693。

任务：将图 5-11 中框起来的部分爬下来，放到一个 TXT 文件中。

步骤：利用 Python 的 BeautifulSoup 模块，将网页内容收纳到容器 soup 中；再在 soup 中搜索所有回复的标识标签；然后将搜索出的内容遍历出来；最后建立一个 TXT 文件，并将遍历结果写入此文件中。

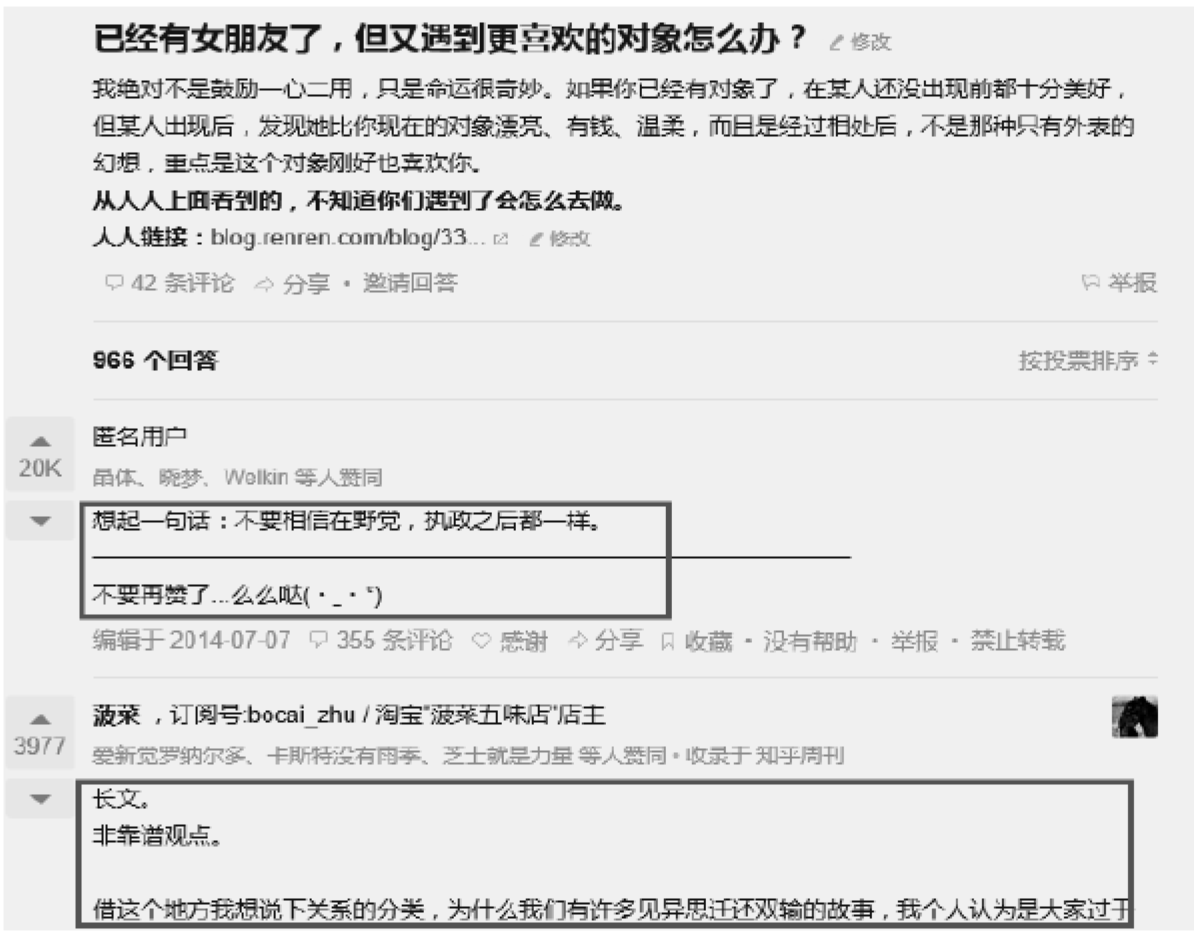


图 5-11 网页截图

(1) 准备工作。先分析网页的源代码，找到回复的标识符。

在网页界面按下 F12 键，如图 5-12 所示。

单击界面上的 B 处(DOM 资源管理器)，再单击 C 处(DOM 元素突出显示)，当把鼠标放到 D、E 区域时，就会发现 A 区域有突出高亮显示。下面的 D、E 区域的每一个 div 标签都对应着上面 A 区域相应的小区域。找到区域 A 中的回复和下面区域 D、E 相对应处，这里提取 D 或者 E 均可，不过 E 比较“干净”，所以就选取 E。其实可以验证，只要是回复部分的内容，它的标签都是 class=“zm-editable-content clearfix”，所以只要找到它的共

性，再把它们全部找出来即可，这是爬取的关键一步。

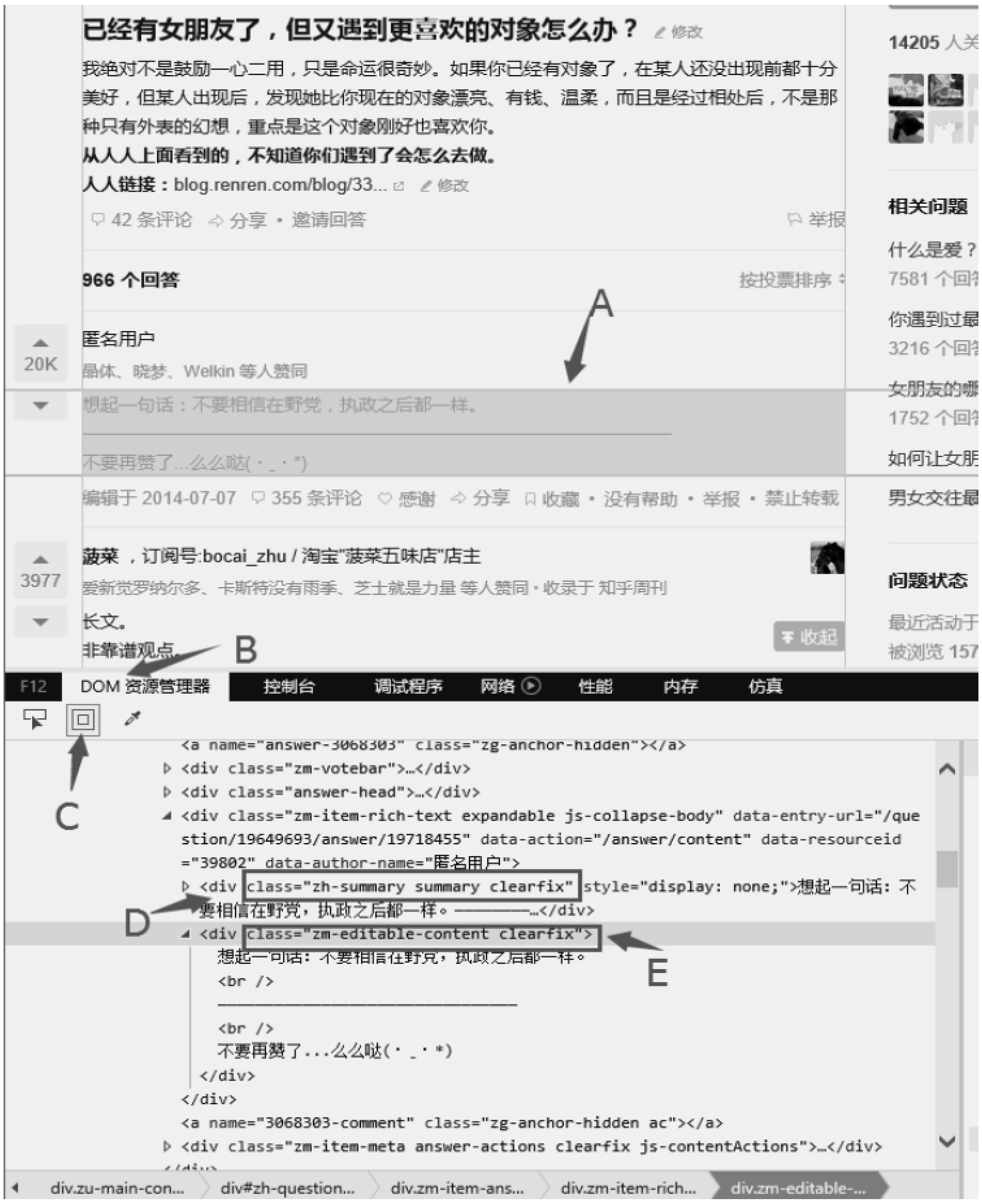


图 5-12　网页源代码截图

(2)　在 Python 中建立 TXT 文件保存爬取的内容，并将网页源代码全部收纳入 soup。

①　在 Anaconda 3 中打开 Spyder，导入相关的模块和库：

```
#encoding:UTF-8
import urllib.request    #在 Python 3.x 中使用 urllib.request 替代了 urllib2
from bs4 import BeautifulSoup  #BeautifulSoup 库最主要的功能是从网页抓取数据
```

②　建立一个 pachong.txt 文件，并允许追加写入：

```
f = open('pachong.txt','a+',encoding='utf-8')
```

其中的 ‘a+’ 表示在文件末尾追加文件；encoding=‘utf-8’ 表示在写入时以 utf-8 编码格式写入，这里最好带上，否则打开 pachong.txt 时会是乱码的。

③ 打开链接网页并爬取网页数据：

```
url="https://www.zhihu.com/question/19649693"   #需要打开的网址
html=urllib.request.urlopen(url).read()          #读取网页数据
soup = BeautifulSoup(html, "lxml")               #将网页数据纳入 soup 中
#print(soup.prettify()) #打印格式，此步可通过打印以监控是否正常。此句可以省略
all = soup.find_all(class_="zm-editable-content clearfix",limit=3)
ALL=str(all).split('</div>')
```

all 行代码：在 soup 中查找标签中的 class="zm-editable-content clearfix"，由于 class 在 Python 中是保留名，所以 class 改写成 class_，即 class_="zm-editable-content clearfix"；limit 表示限制输出的数量，limit=3 表示只输出前三个回复，该参数也可以忽略。

ALL 行代码：是将 all 中的所有内容转化为字符型 str(all)，并将其内容(回复)分割开来，这里使用了字符串分割方法 split，分隔符为'</div>'。

④ 将 ALL 中的内容逐一提取出来，并写入 pachong.txt 文件中：

```
for each in ALL :
    f.write(each+'\n')
```

第二行中使用了'\n'，目的是想每写完一个回复就另起一行，即每条回复另起一行。

⑤ 关闭 pachong.txt 文件，爬虫结束：

```
f.close()
```

这里必须关闭文件，否则会继续占用系统资源。所以每打开一次文件，务必记住用完后关闭该文件，养成良好的编写代码习惯。

至此，爬取的工作基本结束，找到并打开 pachong.txt 文件，验证是否已经爬取了想要的内容。文件的具体保存路径可以在 Spyder 上看到，如图 5-13 所示。

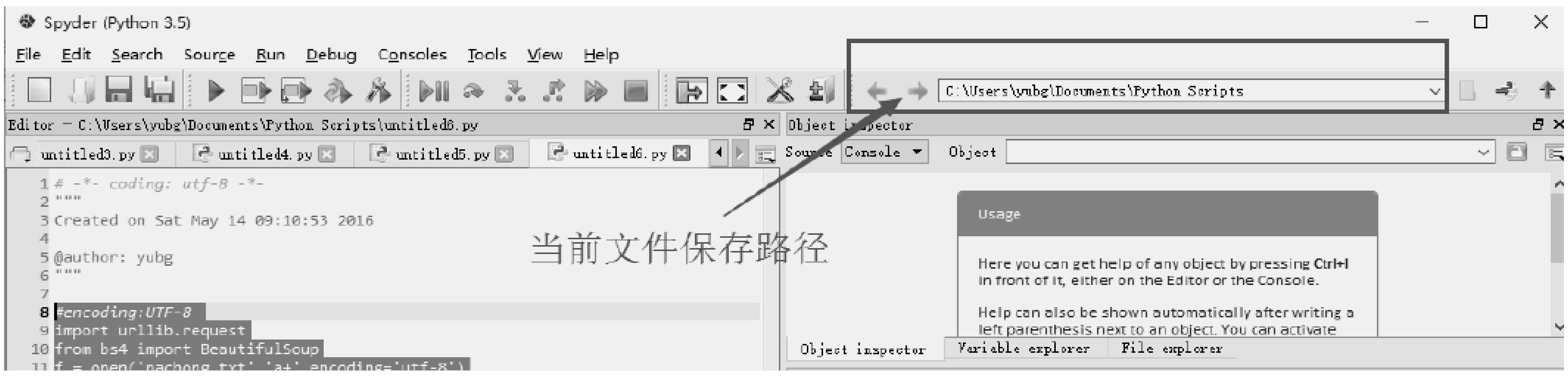


图 5-13 Spyder 保存路径

本爬虫的完整代码如下：

```
#encoding:UTF-8
import urllib.request
from bs4 import BeautifulSoup
```

```
f = open('pachong.txt','a+',encoding='utf-8')
url="https://www.zhihu.com/question/19649693"   #需要打开的网址
html=urllib.request.urlopen(url).read() #读取网页数据
soup = BeautifulSoup(html, "lxml")  #将网页数据纳入 soup 中
#print(soup.prettify()) #打印格式，此步可通过打印以监控是否正常
all = soup.find_all(class_="zm-editable-content clearfix",limit=3)
ALL=str(all).split('</div>')
for each in ALL :
   f.write(each+'\n')
f.close()
```

运行结果如图 5-14 所示。

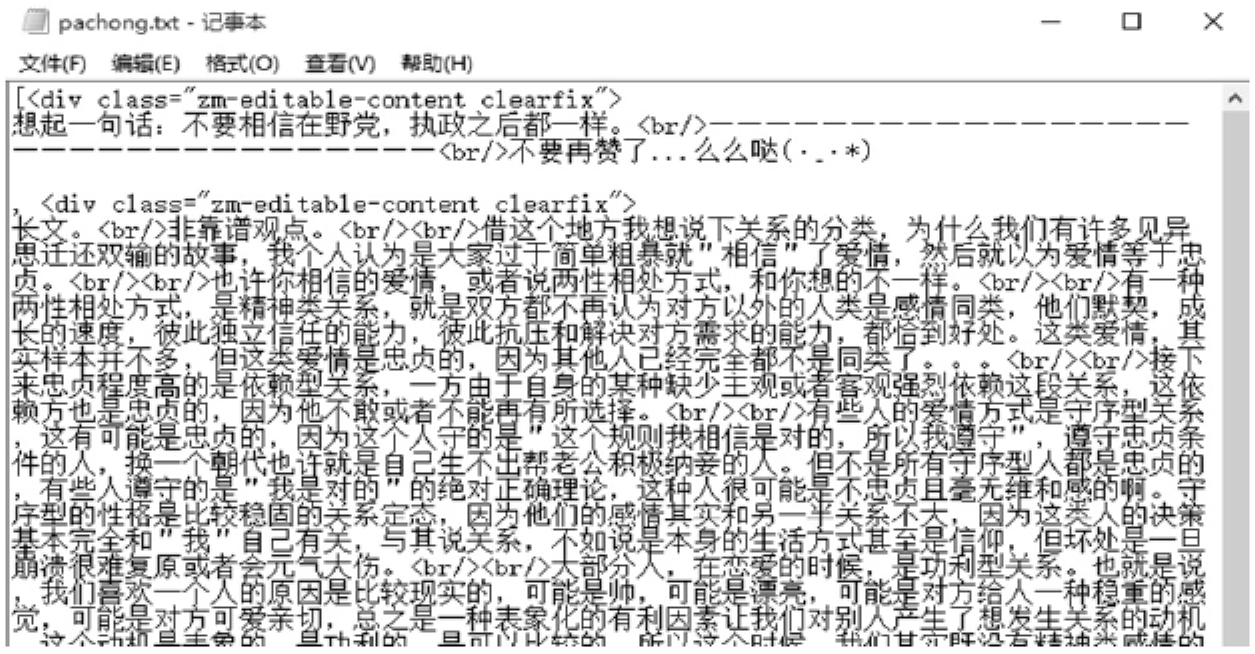


图 5-14　爬取的数据截图

当然，这样还不够美观，还需要继续加工代码。比如每次获取的文本想要知道具体的时间，还想看到这个帖子的标题，并将所有回复按条整理好，如图 5-15 所示。

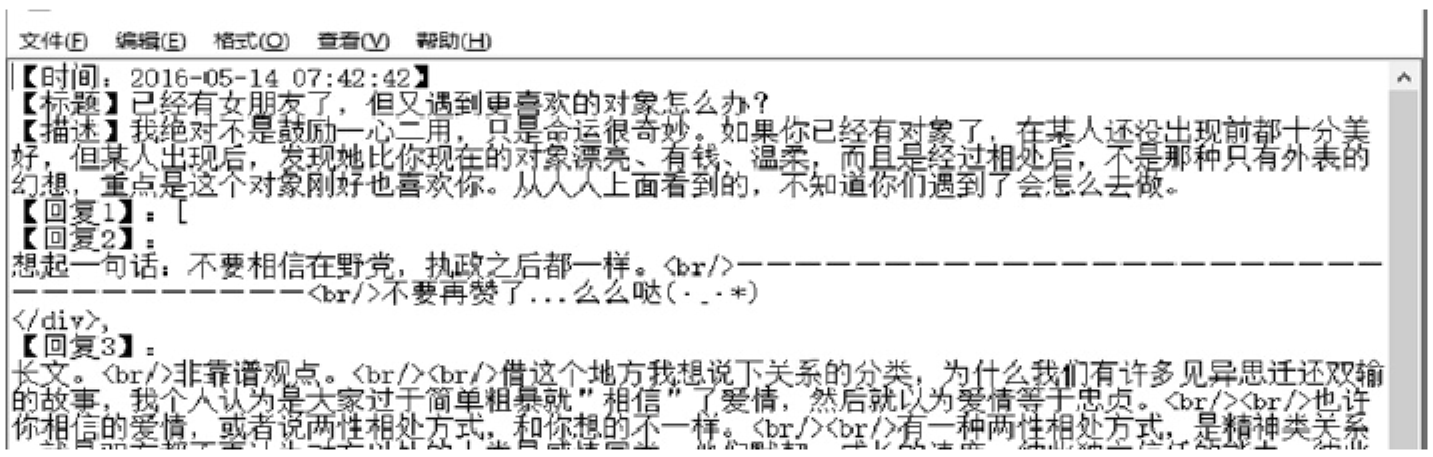


图 5-15　整理后的数据截图

修改后的完整代码如下：

```
import urllib.request
from bs4 import BeautifulSoup
import time
```

```
print("***\n***\n***\n 这是一个爬虫，正在爬取知乎网上的一个内容，请耐心等候：。。。")
f = open('pachong.txt','a+',encoding='utf-8')
end_time=time.strftime('%Y-%m-%d %H:%M:%S',time.localtime(time.time()))
f.write("【时间："+end_time+"】\n【标题】已经有女朋友了，但又遇到更喜欢的对象怎么办？\n【描述】我绝对不是鼓励一心二用，只是命运很奇妙。如果你已经有对象了，在某人还没出现前都十分美好，但某人出现后，发现她比你现在的对象漂亮、有钱、温柔，而且是经过相处后，不是那种只有外表的幻想，重点是这个对象刚好也喜欢你。从人人上面看到的，不知道你们遇到了会怎么去做。"+'\n')

url="https://www.zhihu.com/question/19649693"
html=urllib.request.urlopen(url).read()
soup = BeautifulSoup(html, "lxml")
#print(soup.prettify())   #打印格式，本行可以忽略
all = soup.find_all(class_="zm-editable-content clearfix",limit=3)
ALL=str(all).split('</div>')
i=0
for each in ALL :
    i+=1
    f.write('【回复'+str(i)+'】：'+each+'\n')
f.close()
print("***\n***\n***\n 恭喜你，已经完成任务，请你打开文件：pachong.txt 查阅")
```

end_time 行代码调用的是本地当前时间。执行程序结束后，屏幕上显示的情况如图 5-16 所示，下载内容保存在 pachong.txt 中。



图 5-16　程序运行截图

5.10　读 取 文 档

在很多时候，我们需要读取网上 PDF 文档.pdf 里和 Word 文档.docx 里的文本(字符串)，这时我们就需要下载后再打开查阅。是否可以不下载而直接查阅其文档内的文本呢？方法是有的，Python 就擅长此项工作。

1．PDF 文件

目前很多 PDF 解析库都是用 Python 2.x 版本建立的，还没有迁移到 Python 3.x 版本。但是，因为 PDF 比较简单，而且是开源的文档格式，所以有一些给力的 Python 库可以读取 PDF 文件，而且支持 Python 3.x 版本。

PDFMiner3K 就是一个非常好用的库(是 PDFMiner 的 Python 3.x 移植版)。它非常灵活，可以通过命令行使用，也可以整合到代码中。还可以处理不同的语言编码，而且对网络文件的处理也非常方便。

在 https://pypi.python.org/pypi/pdfminer3k 网址可以下载这个模块的源文件，解压并用下面的命令安装：

```
spython setup.py install
```

下面的例子可以把任意的 PDF 读成字符串，然后用 StringIO 转换成文件对象：

```
from pdfminer.pdfinterp import PDFResourceManager, process_pdf
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from io import StringIO
from urllib.request import urlopen
def readPDF(pdfFile):
    rsrcmgr = PDFResourceManager()
    retstr = StringIO()
    laparams = LAParams()
    device = TextConverter(rsrcmgr, retstr, laparams=laparams)

    process_pdf(rsrcmgr, device, pdfFile)
    device.close()

    content = retstr.getvalue()
    retstr.close()
    return content

    pdfFile =
urlopen("http://pythonscraping.com/pages/warandpeace/chapter1.pdf")
```

```
outputString = readPDF(pdfFile)
print(outputString)
pdfFile.close()
```

readPDF 函数的最大好处是，如果你的 PDF 文件在电脑里，你就可以直接把 urlopen 返回的对象 pdfFile 替换成普通的 open()文件对象：

```
pdfFile = open("d:\11.pdf",'rb')
```

输出结果可能不是很完美，尤其是当 PDF 里有图片、各式各样的文本格式，或者带有表格的数据图时。但是，对大多数的纯文本文档的 PDF 而言，其输出结果还是可以的。

对于网络或者本地 PDF 文档的读取，可以合二为一，对上面的代码稍加修改即可：

```
# -*- coding: utf-8 -*-
from pdfminer.pdfinterp import PDFResourceManager, process_pdf
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from io import StringIO
from urllib.request import urlopen

def readPDF(pdfFile):
    rsrcmgr = PDFResourceManager()
    retstr = StringIO()
    laparams = LAParams()
    device = TextConverter(rsrcmgr, retstr, laparams=laparams)

    process_pdf(rsrcmgr, device, pdfFile)
    device.close()

    content = retstr.getvalue()
    retstr.close()
    return content

path = input('请输入你要查询的具体的网址或路径。如：c:\\11.pdf)：')
if 'http' in path:
    pdfFile = urlopen(path)  #网上搜索
else:
    pdfFile = open(path,'rb')  #本地机器搜索
outputString = readPDF(pdfFile)
print(outputString)
pdfFile.close()
```

以上是在知道具体文件名及路径时，可以打开 PDF 文件并获取其中的文本或者字符串，但这并没有什么太大的实际意义。如果按给定的关键词在本地指定的路径(文件夹)下

查找所有的 PDF，并将找到的文件给出列表，那将有很强的实际意义。下面的代码则实现了这个功能：

```
# -*- coding: utf-8 -*-
'''
Python 3 需要下载安装 pdfminer3k
'''

from pdfminer.pdfinterp import PDFResourceManager, process_pdf
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from io import StringIO
import os
def PdfRead(path):
    content=""
    with open(path,'rb') as pdfFile:
        rsrcmgr = PDFResourceManager()
        retstr = StringIO()
        laparams = LAParams()
        device = TextConverter(rsrcmgr, retstr, laparams=laparams)
        process_pdf(rsrcmgr, device, pdfFile)
        device.close()
        content = retstr.getvalue()
        retstr.close()
    return content
def find_key(path,key):
    for i in os.walk(path):
        for j in i[2]:
            i1=j.find('.pdf')
            if (i1!=-1 and i1==len(j)-4):
                pp=i[0].replace('\\','/')
                try:
                    if PdfRead(pp+'/'+j).find(key)!=-1:
                        print(pp+'/'+j)
                except Exception as e:
                    with open(cur+'/err.log','a') as err:
                        err.write(str(e)+'\n')
if __name__=='__main__':
    find_key(r'C:\Users\yubg\Documents\Python Scripts','Best seats in the
house')
```

上述代码在 C:\Users\yubg\Documents\Python Scripts 下搜索包含有关键词句“Best seats

in the house”的 PDF 文件。搜索结果显示找到了一条记录 C:/Users/yubg/Documents/Python Scripts/A.pdf。如下所示：

```
runfile('C:/Users/yubg/Desktop/pdf.py', wdir='C:/Users/yubg/Desktop')
WARNING:pdfminer.layout:Too many boxes (109) to group, skipping.
C:/Users/yubg/Documents/Python Scripts/A.pdf
```

2. Word 的.docx 文件

大约在 2008 年以前，微软 Office 产品中 Word 使用.doc 文件格式。这种二进制格式很难读取，而且能够读取 Word 格式的软件很少。为了跟上时代，让自己的软件能够符合主流软件的标准，微软决定使用 Open Office 的类 XML 格式标准，此后新版 Word 文件才与其他文字处理软件兼容，这个格式就是.docx。

现在有 Word 文档 http://pythonscraping.com/pags/AWordDocument.docx，我们读取其文档内容：

```
from zipfile import ZipFile
from urllib.request import urlopen
from io import BytesIO
import re
wordFile = urlopen("http://pythonscraping.com/pages/AWordDocument.docx").read()
#wordFile = urlopen("file:///C:/Users/yubg/Desktop/2.docx").read()
wordFile = BytesIO(wordFile)
document = ZipFile(wordFile)
xml_content = document.read('word/document.xml')

s=re.findall(r'<w:t>(.*?)</w:t>',xml_content.decode('utf-8'))
for i in s:
    print(i)

#获取内容如下：
A Word Document on a Website
This is a Word document, full of content that you want very much.
Unfortunately, it’s difficult to access because I’m putting it on my
website as a .
docx
```

为了便于在本地机器上通过给出关键词和指定的路径(文件夹)即可搜索出该文件夹下所有包含要查找内容的.docx 文档，并把含有搜索内容的文件列出，这里给出实现此功能的具体代码：

```
# -*- coding: utf-8 -*-
import os
```

```
from win32com import client as wc

cur=os.getcwd()+'/temp'
if not os.path.exists(cur):
    os.mkdir(cur)

def DocRead(file):
    doc=wc.Dispatch('Word.Application').Documents.Open(file)
    doc.SaveAs(cur+'/txt',4)
    doc.Close()
    txt=''
    with open(cur+'/txt.txt','r') as f:
        txt=f.read()
    os.remove(cur+'/txt.txt')
    return txt

def research(path,key):
    '''
    本代码的功能：
    主要用于某文件夹下搜索 Word 文件内的内容。
    如想搜索关键词为“大数据分析报告的主要内容如下：”这个片段
    '''
    n=0
    for i in os.walk(path):
        for j in i[2]:
            i1=j.find('.doc')
            i2=j.find('.docx')
            if (i1!=-1 and i1==len(j)-4) or (i2!=-1 and i2==len(j)-5):
                pp=i[0].replace('\\','/')
                # print(pp+'/'+j)
                try:
                    if DocRead(pp+'/'+j).find(key)!=-1:
                        print(pp+'/'+j)
                        n+=1
                except Exception as e:
                    with open(cur+'/err.log','a') as err:
                        err.write(str(e)+'\n')
    if not n:
        print('没有找到你要的内容！')
    else:
        print('找到%d 条内容'%n)

def main():
```

```
    path=input('输入查找根路径(如: F:\): ')
    key=input('输入查询关键字: ')
    research(path,key)

if __name__=='__main__':
    main()
```

在本地机中，C:\Users\yubg\OneDrive\other 下有某个文件中含有“中北大学单位内各部门的办公文档、卷宗、图片、视频等数据”的关键词句。执行上面的程序，输入查找路径和关键词句，查询结果如图 5-17 所示。



图 5-17　查询结果

本 章 小 结

本章主要学习了对文件的读取和存储，以及如何使用 with 语句；要求掌握计算精度的处理方法，以及正则表达式的使用，并对矩阵的基本运算有所了解。

练　　习

(1) 求矩阵 A 的伴随矩阵。AA*=|A|I，A=np.mat([[1,2,3],[4,5,6],[7,8,9]])。

(2) 判断 RE 字符串是否全部为小写。RE=‘Alphabet3’。

(3) 写一个以正则表达式匹配 IP 地址的代码。

(4) 简单的网络爬虫。要求获取页面 http://t.cn/RVVe68d 中的省市名称、作物名称和面积这三个值，并打印出来。(在项目中，需要使用 beautifulsoup4，所以要首先安装 beautifulsoup4，并且爬取网页的时候使用 requests 模块)

参 考 文 献

[1] 齐伟. 跟老齐学 Python 从入门到精通[M]. 北京：电子工业出版社，2016.

[2] 杨佩璐，宋强 等. Python 宝典[M]. 北京：电子工业出版社，2014.

[3] 董付国. Python 程序设计[M]. 北京：清华大学出版社，2015.

[4] 张良均，王路，谭立云，苏剑林 等. Python 数据分析与挖掘实战[M]. 北京：机械工业出版社，2016.

[5] [印]Ivan Idris. Python 数据分析基础教程 NumPy 学习指南[M]. 2 版. 张驭宇 译. 北京：人民邮电出版社，2014.

[6] [美]Mark Lutz. Python 编程[M]. 4 版. 邹晓，瞿乔，任发科 等译. 北京：中国电力出版社，2015.

[7] [美]Ryan Mitchell. Python 网络数据采集[M]. 陶俊杰，陈晓莉 等译. 北京：人民邮电出版社，2016.